AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

1 OF 9
AD
A069036

LEVEL II

42

Technical Report: NAVTRAEQUIPCEN 77-C-0162-2

GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM

Gene D. Barber, Mary Hicklin, Catherine Meyn,
Jack E. Porter, Gail K. Slemon
Logicon, Inc.
Tactical and Training Systems Division
P. O. Box 80158
San Diego, California 92138

April 1979

INTERIM REPORT 23 SEPTEMBER 1977 - SEPTEMBER 1978

NAVAL TRAINING EQUIPMENT CENTER
ORLANDO, FLORIDA 32813

NAVTRAEQUIPCEN 77-C-0162-2

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER NAVTRAEQUIPCEN 77-C-0162-2	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.		5. TYPE OF REPORT & PERIOD COVERED Interim Report, 23 Sep 1977 - 25 Sep 1978
		6. PERFORMING ORG. REPORT NUMBER 5581-0005
7. AUTHOR(s) Gene D. Barber, Mary Hicklin, Catherine Meyn, Jack E. Porter, Gail K. Slemon		8. CONTRACT OR GRANT NUMBER(s) N61339-77-C-0162
9. PERFORMING ORGANIZATION NAME AND ADDRESS Logicon, Inc., Tactical & Training Systems Div. P.O. Box 80158 San Diego, CA 92138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 7796-1P1
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Training Equipment Center Orlando, Florida 32813		12. REPORT DATE April 1979
		13. NUMBER OF PAGES 784
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) 787p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive Training
Controller Training System
Digital Speech Record/Playback
Ground Controlled Approach
Performance Measurement
Speech Synthesis
Speech Recognition
Speech Understanding
Voice Data Collection

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes the design of the hardware and software system which will support the training envisioned for the experimental prototype Ground Controlled Approach Controller Training System (GCA-CTS). Included as appendices are a facilities report, system interface drawings, trainee and instructor panel drawings, and programming variable definitions and file structures.

411171

LB

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont)

TABLE OF CONTENTS (Cont)

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont)

LIST OF TABLES

LIST OF TABLES (Cont)

SECTION I

INTRODUCTION

Background

The Ground Controlled Approach Controller Training System, the GCA-CTS, will provide basic training in the conduct of ground controlled approaches using simulated precision approach radar (PAR) equipment. The PAR indicator provides aircraft elevation, azimuth, and range information on final approach. The PAR controller's task commences when he or she assumes responsibility for the control of an aircraft after handoff from the pattern controller. That responsibility terminates when the aircraft reaches decision height, although the controller continues to give advisories until the aircraft passes landing threshold or executes a missed approach.

During the approach the controller issues course corrections and glidepath advisories over a voice channel to enable the pilot to effect a safe approach even during periods of low ceiling and visibility, regardless of the NAVAID receiving equipment in the aircraft. A well defined, precise radio terminology (R/T) serves as the vehicle for this communication. The controller training problem, therefore, involves teaching the student to interpret the radar display, to determine appropriate corrections and advisories, to communicate this information to the pilot in a standard format and, in addition, to coordinate with other air traffic control (ATC) personnel.

The GCA-CTS

The GCA-CTS will be an experimental prototype training system (as contrasted with a training device) designed to provide basic training in GCA procedures. It will be designed to ensure that competent trainees master the basic skills within the five-day time frame allowed in the present course. Because of the many features of the GCA-CTS, students are expected to complete basic training in significantly less than five days, and enrichment exercises will be provided. The GCA-CTS will provide automated, individualized instruction with objective performance assessment and numerous instructional aids including illustrated texts, computer-aided instruction, adaptive problem selection, detailed performance summaries, and annotated replays.

The GCA-CTS will benefit the students in other ways as well. It will relieve them of pseudo-pilot duties which do not contribute to the acquisition of controller skills but which must be performed with the current training device. It will also provide the faster students with opportunities for the acquisition of advanced skills, since post-graduate training will be available for those students who complete the basic course quickly. Another major advantage of the system is that it will relieve the instructor of many of those routine duties which encroach on his or her training management time.

The existing laboratory GCA-CTS demonstrated the feasibility of a GCA controller training system in which the student's verbal behavior is automatically monitored and scored with the aid of commercially available speech recognition hardware. In addition, it demonstrated that a syllabus could be constructed and that automated adaptive training of the task with objective performance measurement was possible.

The experimental prototype GCA-CTS will embody all the lessons learned in the laboratory system and will incorporate additional sophisticated training techniques. It will be designed for motivated and responsible students. The intent is to produce a system that will provide a challenging and interesting learning environment for the individual student. This requires a course adaptively tailored to meet individual needs, with clearly defined objectives which are challenging but attainable.

Focus of the System Configuration Report

The Training/Functional Design Report, delivered in February of this year, described the behavioral objectives which the student must attain in order to pass the training course. It then described the course syllabus which was designed to meet the behavioral obejctives. Finally, it detailed the functional requirements of a system which could support the training envisioned for the experimental prototype GCA-CTS.

The present document describes the hardware and software which will satisfy these functional requirements. Its purpose is to guide the implementation effort. As a design document, the material presented herein is almost certain to change. Nonetheless, it is presented in a formal way to give the government cognizance of system operation prior to implementation. The report consists of five sections and a set of appendixes. Section 2 describes the hardware environment. Section 3 details the design of the special purpose devices required by the GCA-CTS. Section 4 covers the software environment which will support the applications routines. Section 5 shows the design of these applications routines.

A Facilities Report is provided as Appendix A of this document. Working drawings of the hardware logic diagrams for the Logicon-built devices are included as Appendixes B, C, and D. Internal data structures are shown in Appendix E, and file structures in Appendix F. Appendix G describes the format of the software logic diagrams included in this report.

Section II

THE HARDWARE ENVIRONMENT

Overview

The operational hardware is combined into three assemblies. Each assembly will normally be at a separate location. The main assembly is the system controller, configured as a double-bay cabinet 46 inches wide by 32 inches deep by 70 inches high. A second assembly is the trainee station, consisting of a desk holding several computer peripheral devices. The third assembly is the instructor station, also a desk holding several peripheral devices. The stations may be located up to 100 cable feet from the system controller in different directions. The system controller contains two central processing units, disk and diskette storage, and audio input and output units. The stations provide audio, visual, and manual interfaces to facilitate training and instructor monitoring.

Figure 1 presents a hardware block diagram. Figure 2 indicates equipment grouping and cabling. Appendix A presents the installation floor plan.

Most of the equipment is available commercially. Logicon, however, has designed four items and details of these are presented in Section 3 and in Appendixes B, C, and D.

Particulars of each of the three assemblies are presented in the following paragraphs. These are followed by a discussion of various Integrated Logistic Support (ILS) topics.

Appendix A, the Trainer Facilities Report, provides specifications and information to facilitate advance preparation for an easy installation of the system. That information will not be repeated below.

The System Controller

This is contained in a double-bay cabinet, Data General Corporation (DGC) Model 1012L. This cabinet and most of its contents are government furnished equipment and, as such, should need no lengthy description. These GFE items include:

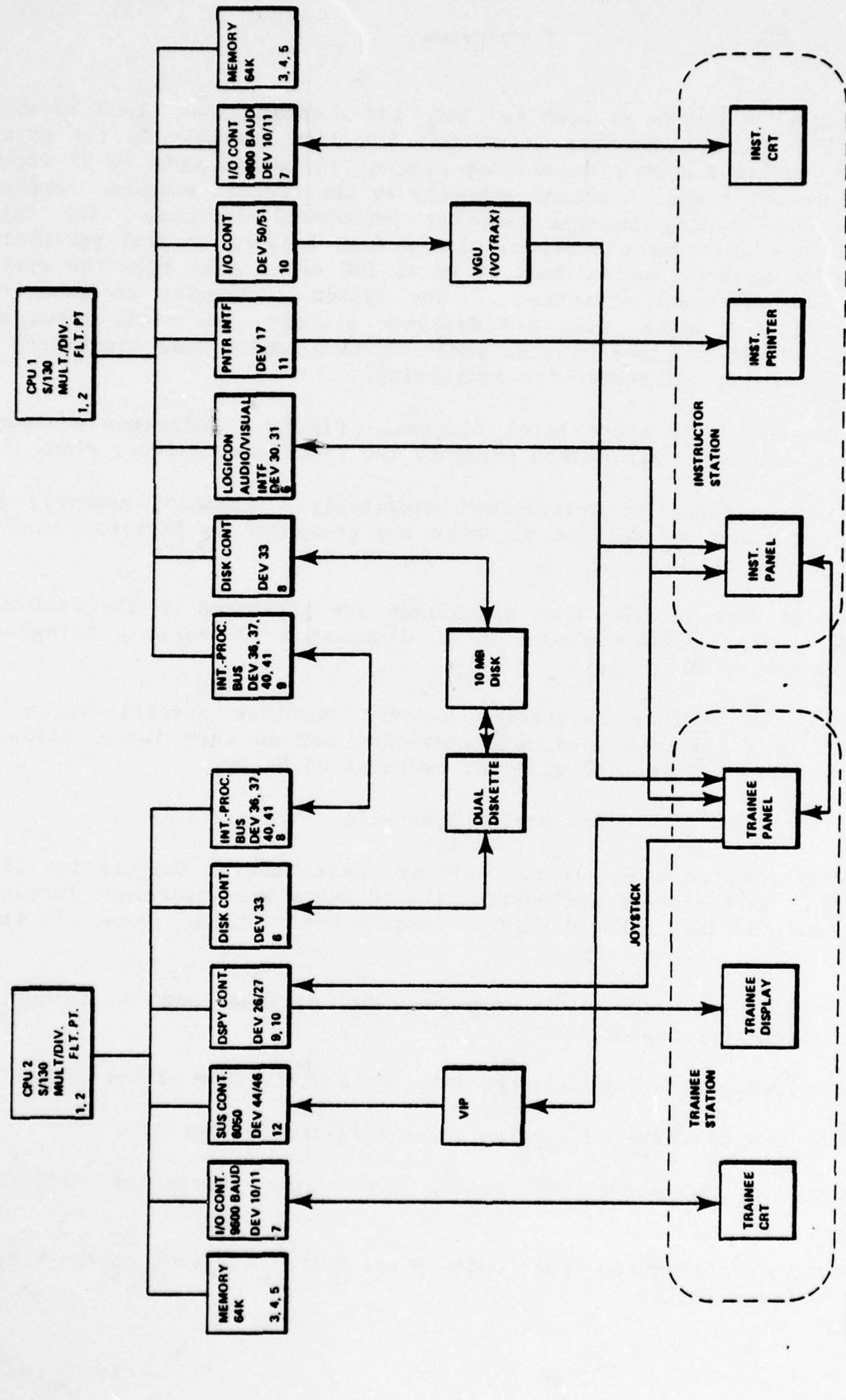
2 each — Eclipse S/130 CPU with various options and a connecting inter-processor bus.

1 each — 10 Megabyte Disk Storage Unit, accessible from either CPU.

1 each — Dual Diskette Unit, also accessible from either CPU.

In addition to the above GFE items, the system controller will also contain:

1 each — Voice Generation Unit made by the Votrax Division of the Federal Screw Works.



NOTES:
1. NUMBERS IN LOWER LEFT CORNERS ARE SLOT LOCATIONS IN CPU CABINET
2. CPU 2 SLOT 11 HAS TERMINATOR ONLY
   CPU 1 SLOT 12 HAS TERMINATOR ONLY.

Figure 1. GCA-CTS Hardware Block Diagram

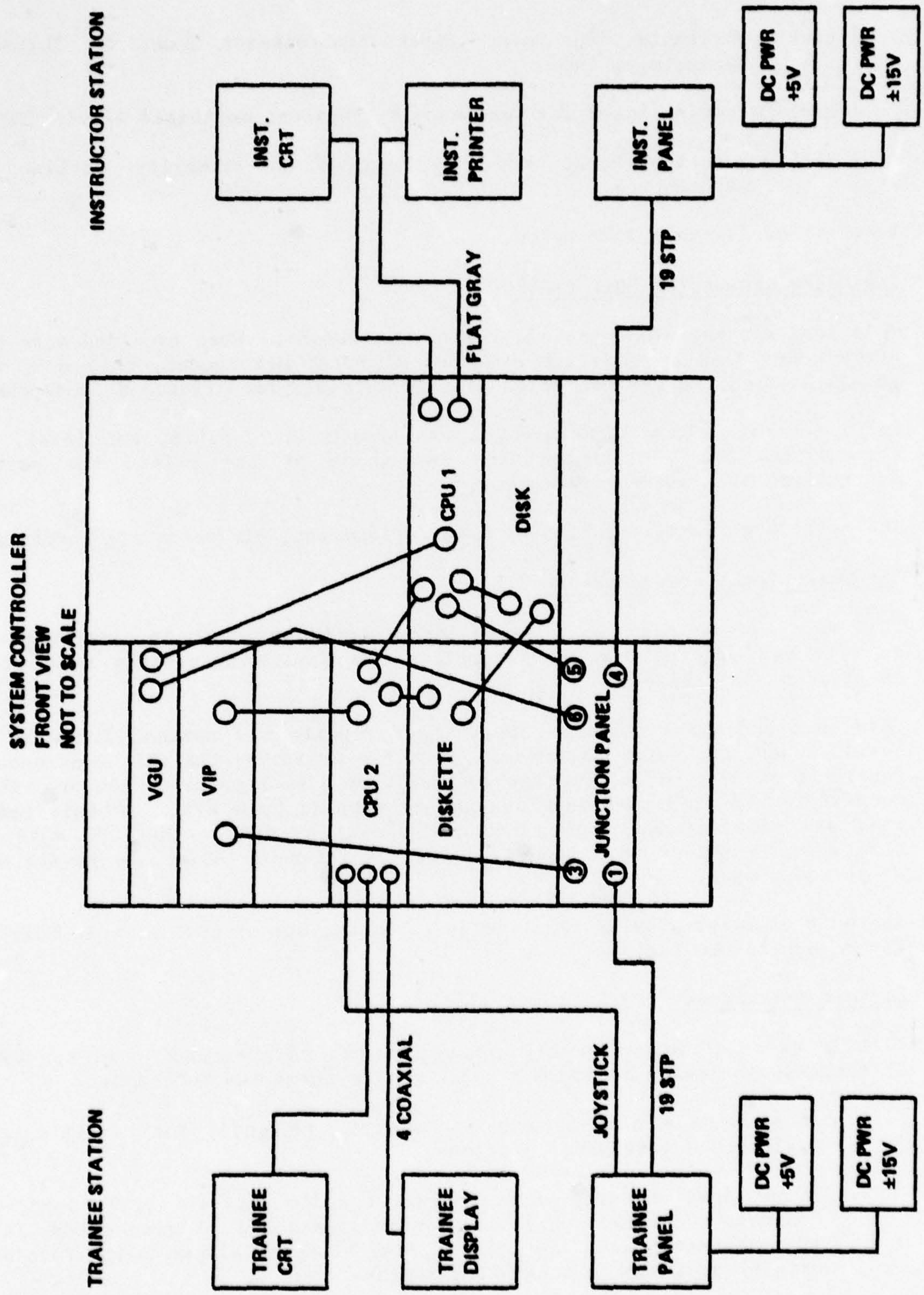


Figure 2. GCA-CTS Major Cabling

1 each — Threshold 500 Voice Input Preprocessor made by Threshold Technology, Inc.

1 each — Special Interface Card made by Logicon; contained within CPU1.

1 each — Junction Panel made by Logicon to simplify cabling and maintenance.

These items are described below.

The Voice Generation Unit (VGU)

This is a rack-mounted unit, 3.5 inches in height. When provided with properly coded data at the rate of about 15 bytes per second, this unit will generate easily understood voice signals suitable for driving a loudspeaker.

Panel controls allow adjustment of the speech rate, pitch, and level. In this system the VGU can provide the voice of the pilot, the pattern controller, or a pseudo instructor.

The unit is normally repaired by card replacement; six cards are involved.

The Voice Input Preprocessor (VIP)

This is a rack-mounted unit, seven inches in height. It is slide mounted, and its two printed circuit assemblies are easily accessible when it is extended on the slides.

This unit accepts a balanced audio input signal at a nominal 2.5 volt RMS level. When this input is present (from the trainee), the unit continuously generates 32 bits of information available on a back panel connector. These constitute two 16-bit words which may be sampled by a CPU. In this system they are sampled about 450 times per second by CPU2. The CPU makes an assessment, based on this input and various internal tables, as to the word or phrase spoken.

The unit will normally be repaired by replacing one or both circuit boards s for return to the factory.

Logicon Interfaces

This is an electronic assembly constructed as an interface card for CPU1. It responds to device codes 30 and 31. It performs two functions:

1. It acts as a link between the lights, switches, and alarms in the trainee and instructor stations.

2. It provides encoding and decoding of audio data to 16-bit words and controls the data channel storing and accessing of those data. This, in conjunction with the digital disk storage, allows audio recording with instant random access for replay.

This unit is discussed further in Section 3.

The Junction Panel

This is discussed in Section 3.

The Trainee Station

Appendix A presents physical information on this unit. It is a desk which holds three major components:

1. A Megatek MG552 graphic CRT display.

2. A Data General Corporation Model 6053 Video Display Terminal with keyboard. This is GFE.

3. A Trainee Panel designed by Logicon to provide lights, buttons, sounds, and a joystick for a trainee interface.

Below the desk surface, near the rear, will be a power distribution strip and two DC power supplies for the Trainee Panel.

The Graphic Display

The purpose of this display is to present graphic, simulated-radar images to the student. It presents a 21-inch display with a resolution of 4096 on either axis. It is controlled by four video signals from a vector generator card within CPU2 in the system controller.

The Video Display Terminal

This unit operates at 9600 baud and functions as the normal console input/output device for CPU2.

The Trainee Panel

This unit is designed by Logicon to simulate actual equipment to provide realism to the training. It is positioned to the right of the radar-simulating display. It is a box 17 inches wide by 12 inches deep by 11 inches high. The front panel slopes backwards at 15 degrees from the vertical.

The unit contains Logicon-designed circuitry plus circuit cards for the Megatek joystick and the Threshold Technology VIP preamplifier. Panel lights, switches, and audible alarm are programmed as device 30 of CPU1.

Audio circuitry allows microphone input and headset or speaker output for a variety of audio sources/destinations including the instructor station, the VGU, the VIP, and the device 31 recording/random playback function.

Details are presented in Section 3.

DC power for this unit comes from supplies mounted below the desk top on the backside of a modesty panel.

The Instructor Station

Appendix A presents physical information on this unit. It is generally similar to the trainee station; two of the major equipments, however, are different. The major equipments are:

1. A Data General Corporation Model 6053 Video Display Terminal with keyboard as used in the trainee station. This is GFE.

2. A Tally Model 1602 serial character printer.

3. An Instructor Panel designed by Logicon to permit audio communication with the trainee and exercise monitoring.

Below the rear desk surface is a power distribution strip and two DC power supplies for the instructor panel.

The Serial Printer

This printer is Tally Model T1602. It quietly prints 160 characters per second, bidirectionally on an original and up to four carbon copies. Up to 132 characters may be printed per line.

The printer is controlled by bytes, transmitted on eight parallel lines from a controller in CPU1.

The printer will be used operationally to provide trainee performance evaluations, diagnostic messages, summary reports, etc. It will provide listings during software development.

The Instructor Panel

This unit functions as an intercom to the student. It also allows audible monitoring of the various voice sources in the system. It is designed by Logicon and is housed in a box 17 inches wide by 12 inches deep and 7 inches high.

Details are presented in Section 3.

Maintainability Considerations

In the paragraphs below are presented some considerations on maintainability.

Equipment Reliability

Data on mean-time-to-failure has been requested from the five major vendors. Data has been obtained in various ways. It is presented as table 1.

Values for Logicon-produced devices have been calculated from MIL Handbook 217 and are shown in table 2. When failures from both tables are added, a mean-time-to-failure of 304 hours is obtained.

TABLE 1. VENDOR PRODUCTS RELIABILITY

Vendor	Quan.	MTTF Unit	MTTF Total	F/ 10^6Hr	Remarks
Data General Corp.					
8611K S/130	2	10,374	5,187	192	MIL Hdbk. 217 Rev. B
8613 FLT PT	2	1.37M	.69M	1	All
4007 I/F Card	3	249K	83K	12	
4008 I/F Card	2	2.2M	1.1M	1	
4010 I/F Card	3	973K	324K	3	
4023 I/F Card	2	15M	7.5M	1	
4029 I/F Card	1	5.8M	5.8M	1	
6045 Disk	1	9.9K	9.9K	101	
6030 Diskette	1	17.8K	17.8K	56	
6053 AA Dasher	2	29.4K	14.7K	68	
4240 IPB	2	50.3K	25.1K	40	
DGC Total				476	
Megatek		2000	2000	500	Estimated by Megatek
Votrax	1	9K	9K	111	MIL Hdbk. 217 Rev. B
Threshold Technology					
Pwr Sup	1	2 yr		57	Observed
Bd 4501	1	10.2 mo		133	
Bd 4502	1	11.2 mo		122	
I/F 6050	1	11.4 mo		120	
Total Treshold Tech.				432	
Tally	1	1753		570	Observed as of Aug 77
Total 5 Vendors			478	2,089	

MTTF per unit and total refer to mean-time-to-failure per hour.
1 mo = 730 hr.
1 yr = 8760 hr.
M = $1*10^6$
K = $1*10^3$

TABLE 2. LOGICON PRODUCT RELIABILITY

	MTTF Unit	MTTF Total	F/ 10^6Hr	Remarks
Logicon				
I/F Dev 30, 31				
Std Panel			1,110	
Inst Panel				
Pwr Sup $\pm$15v 2	35K	17.5K	57	MIL Hdbk. 217 Rev. B
$\pm$ 5v 2	68K	34.0K	29	MIL Hdbk. 217 Rev. B
Total Logicon			1,196	

Maintenance Manuals

An evaluation of vendors' maintenance manuals is in process. Current evaluations are as follows:

- Data General Corp. Adequate
- Tally (Printer) Adequate
- Votrax (VGU) Adequate
- Megatek In process for many months. Reportedly, Logicon will receive an early copy in October, 1978.
- Threshold Technology (VIP) None available and none planned. Logicon is getting the cost estimate for a manual. Actually, only two plug-in printed circuit assemblies and a power supply are involved. Tentative feeling is that board swapping is more feasible than board repair due to specialized nature of the circuit.

Logicon will supply an adequate maintenance manual for Logicon-designed items as well as system trouble localizing suggestions.

Maintenance Training Courses

A survey of vendor-provided training courses is being made. Results as of this writing are indicated in table 3. No recommendation that these courses should be taken is implied. The adequacy of the training depends on the background of the student. A medium level of competency with knowledge of digital integrated circuits is assumed. If courses are taken, it is desirable that Data General Corporation courses be attended first.

TABLE 3. VENDOR MAINTENANCE TRAINING COURSES

Vendor Course	Duration	Cost	Remarks
DGC			
H100 Computer Hardware Fundamentals	1 wk	$450/person	All courses at Westboro, Massachusetts
H104 Eclipse S Series Familiarization	2 wks	$900/person	Adequate for board level maintenance
H304 Eclipse S130 Maintenance	1 wk	$450/person	Also needed for chip level maintenance
H202 Diskette Familiarization	3 days	$270/person	
H203 Disk (6045) Familiarization	1 wk	$450/person	
Tally			
S1000 Inplant	3 days	$225/person	At Kent, Washington
Onsite		$250/day	Plus expenses for instructor, several students allowed. Portal-portal.
Votrax		Pending	
Megatek			
Inplant	2 days	$320/day	1 or 2 students at San Diego, California
Onsite		$320/day	Plus expenses for instructor, Portal-portal.
Threshold Technology	1 wk	$600/person	At Delran, New Jersey. Assumes user has spare boards plus diagnostic unit. Board replacement.

Maintenance Agreements

The five major vendors have been contacted regarding maintenance agreements. The general nature of their standard agreements is as follows:

- Response time of 4 to 24 hours.
- Contract holders have priority over other users.
- Fee is about one percent per month. Contracts are normally one-year maximum; longer contracts perhaps with fee change.
- Additional charges to validate equipment not under warranty at the time.
- Optional supervision of packing methods before a move.
- At new site, after a move, there are usually revalidation and possible repair, both on a time and material basis, prior to resuming maintenance agreement.
- Possible cost adjustment at new site if mileage is excessive.
- Some vendors provide preventative maintenance as part of the agreement.

Table 4 presents some particulars for the five vendors. All are for service during normal working hours.

TABLE 4. VENDOR MAINTENANCE AGREEMENTS

Vendor	Fee	PM Included	Revalidation After Move	Remarks
Data General Corp.	$654/mo	Yes	$200 Estimate (T&M)	Onsite
Tally	$576/yr	No	Not Required	For 50 to 100 miles. Should be $450/yr at Orlando and Memphis, which are service centers.
Votrax	$32.50/mo	No	Not Required	All suspected faulty equipment must be sent to Troy, Michigan for analysis and repair or replacement. Estimated downtime is 4 to 7 days without spares.
Megatek	$160/mo	No	$1000 Estimate (T&M) Travel	Additional travel expenses for service calls in Orlando for 3 month period, which is too short to employ and train a local service organization.
Threshold Tech.	$792/yr	No	Yes, on T&M	Onsite

1. T&M means time and material charges.
2. Agreements are for service during normal working hours.

SECTION III

SPECIAL HARDWARE

Overview

Logicon is developing a subsystem involving several audio, visual, and switch elements to facilitate communications between trainee, instructor, and computer. There are four assemblies involved and each is discussed in a separate paragraph below.

Functions of this subsystem are as follows:

- To provide an audio intercom between trainee and instructor
- To provide a method of recording five minutes of trainee voice with playback of any portion with an access time of one second maximum
- To allow trainee and/or instructor to hear the above playback or the VGU output
- To allow computer sensing of certain switch positions at the instructor and trainee stations
- To allow computer control of certain lights and an audio alarm at the instructor or trainee station
- To provide indication to the computer of the completion of a trainee's voice input and its approximate level
- To provide indication to the computer of the completion of VGU audio output

This subsystem is being designed and implemented with due regard to reliability, maintainability and other ILS factors.

Interface/Recorder

This unit is constructed on an interface card which is to be installed in a slot of CPU1. The CPU communicates with it as device 30 and 31. The unit is connected via a 100 foot cable to the trainee panel.

Device 30 acts as an interface to allow the CPU to control lights and an audio alarm at the trainee station and a light at the instructor station. It also allows the sensing of switch positions and audio levels.

Device 31 performs the general function of an audio recorder/playback unit s for the trainee's voice. It makes use of a special integrated circuit called a Continuously Variable Slope Delta Modulator (CVSD). This unit is made by Harris Semiconductor as device HC-55516. It will either encode or decode an audio input. The encoded result is 16,000 bits per second. These form 1,000 computer words per second to be stored in CPU memory buffers by

data channel action. This CPU in turn must store the buffers in real time on the ten megabyte disk. On playback, the process is reversed with the CPU having the option of selecting any desired disk sectors for playback.

Logic diagrams for the unit are contained in Appendix B. To reduce the number of wires involved, the device 30 I/O to and from the trainee station is transmitted/received as 32 bit serial sequences.

All analog and digital signals between this unit and the trainee panel are sent differentially on shielded, twisted pairs of wires within the cable.

Device 30 Hardware Description

Figure 3 presents a block diagram of the device 30 circuitry. Appendix B presents detailed logic. The reader is assumed to be familiar with hardware interfacing of Data General Corporation computers.

At the top of figure 3 are two shift register sections, each of 16 bits capacity. A is loaded first by a computer DOA command. Then B is loaded. Following the DOB, the 32 bits are automatically stepped out in serial fashion to control lights on trainee and instructor panels and an audible alarm.

At the center of figure 3 are two shift register sections totaling 32 bits. These are loaded by serial switch data from the trainee panel. Upon reception of 32 bits a done flip-flop is set and an interrupt is generated. The computer can then input the data by a DIA and a DIB command. Normally, data are received only when changes occur. However, the computer can at any time send a start command which sets a busy flip-flop and initiates a 32 bit transmission from the trainee panel.

Device 30 Programming

DOA and DOB Commands

Lights, alarm, etc. will be controlled by two 16 bit words output from the training system controller by a DOA command followed by a DOB command. Both commands shall be used even though a single bit is changed. Bits indicate desired conditions (1 = on, 0 = off). The commands will become effective at the trainee panel approximately 30 microseconds after the DOB command is given. Busy need not be set. Bit significance is given in table 5. All controlled functions are on the trainee panel except DOB, bits 2 and 3. Note that a steady red light occurs if both red and red flashing bits are set. Also, amber and green lights can be on in the same switch if both bits are set.

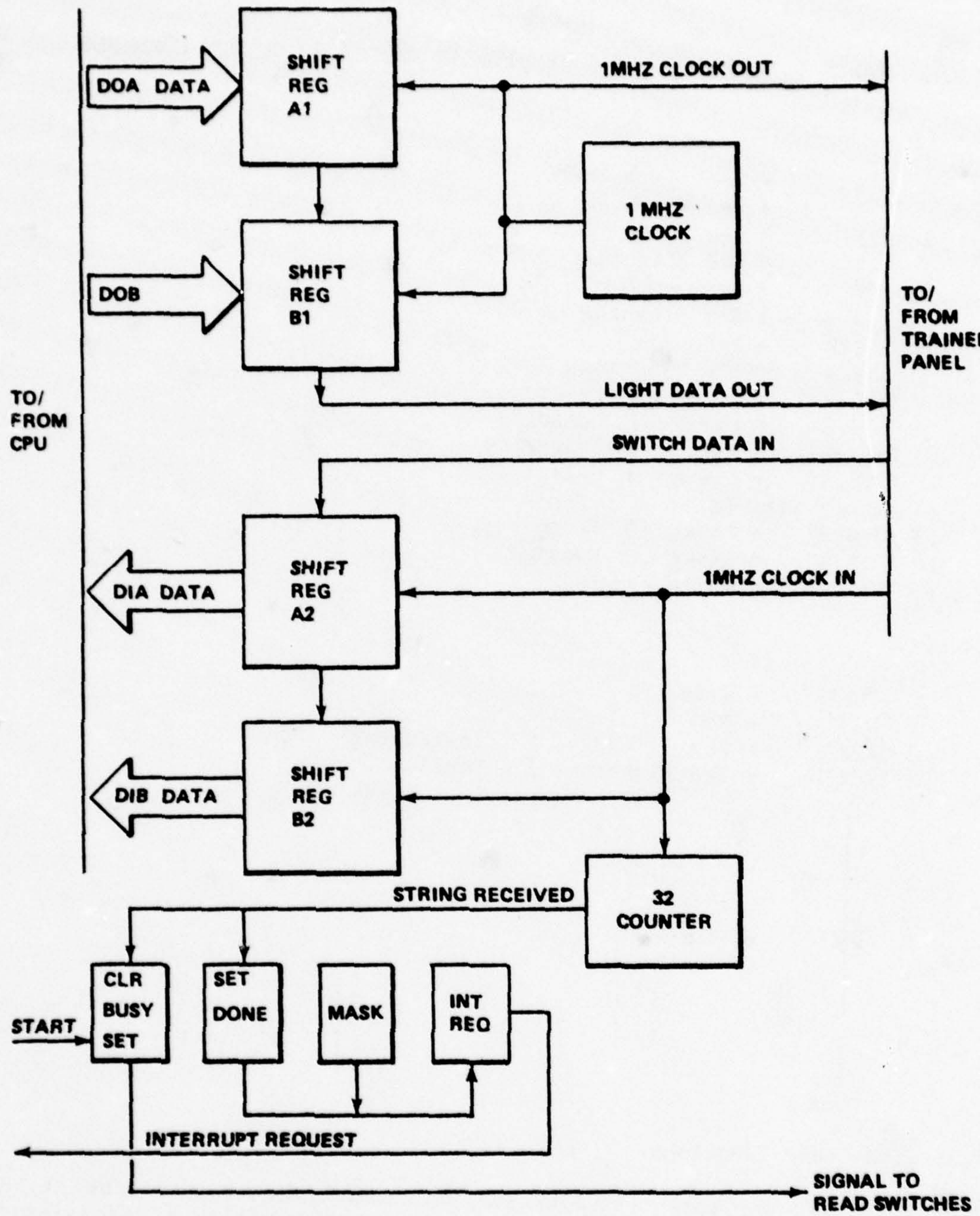


Figure 3. Device 30 Block Diagram

TABLE 5. SIGNIFICANCE OF BITS IN DEVICE 30 OUTPUT COMMANDS

DOA bits:

0	"3"	= Red	
1	"3"	= Red Flashing	
2	"5"	= Red	
3	"5"	= Red Flashing	
4	"7"	= Red	
5	"7"	= Red Flashing	
6	"SUPER"	= Red	
7	"SUPER"	= Red Flashing	
8	"270.8"	= Amber	Frequency Select
9	"270.8"	= Green	
10	"318.8"	= Amber	Frequency Select
11	"318.8"	= Green	
12	Alarm, Audible		
13	"270.8"	= Amber	Frequency Monitor
14	"318.8"	= Amber	
15	"REQUEST"	= White	

DOB bits:

0	"CLEARED"	= Green	
1	"W/O"	= Red	
2	"ICS"	= Red	Instructor Station
3	"ICS"	= Red Flashing	
4–15		Not used	

Start and Clear Functions

Start will set the busy flip-flop and will cause the trainee panel to send 32 bits to the interface. These bits will be sent within 40 microseconds. When they have been received, busy will be cleared and done will be set. This will cause an interrupt unless interrupt disable bit 3 is effective as the result of a MSKO command. The interrupt routine should clear the done flip-flop by a clear command or an I/O reset. The student's panel will automatically send a stream of 32 bits without a start command when certain events occur at the panel. These likewise will set a done flip-flop as discussed above. Events which will initiate the 32 bit transfer are a change in any of the monitored functions, DIA bits.

DIA and DIB Commands

These commands are to be used only to read the 32 bits received when the done flip-flop is set. These commands should be used in tandem, the last one may contain the clear function. The meaning of these 32 bits is shown in table 6. A change in either direction of DIA bits 0 through 12 causes a transmission sequence. Changes in DIB bits do not initiate a transmission.

TABLE 6. SIGNIFICANCE OF BITS IN DEVICE 30 INPUT COMMANDS

DIA bits (1 = condition shown)

0	"3" depressed
1	"5" depressed
2	"7" depressed
3	"SUPER" depressed
4	"270.8" depressed
5	"318.8" depressed
6	"270.8 monitor" depressed
7	"318.8 monitor" depressed
8	"REQUEST" depressed (momentary button)
9	FOOT SWITCH depressed (momentary)
10	VOTRAX active within past 1 second
11	Student voice active within past .25 second
12	Instructor's "ICS" mode set
13, 14, 15	Spare

DIB bits

0, 1, 2, 3	4 bit representation of peak student voice level since last reset. Bit 0 is most significant bit Auto reset occurs about 0.1 second after DIA bit 11 falls.
4 . . . 15	not used

Device 31 Hardware Description

Figure 4 presents a block diagram and timing for the device. Buffers A and B are in the S/130 memory and are of 256 to 2048 words in length. The timing diagram was drawn to show disk timing for the 256 word buffers.

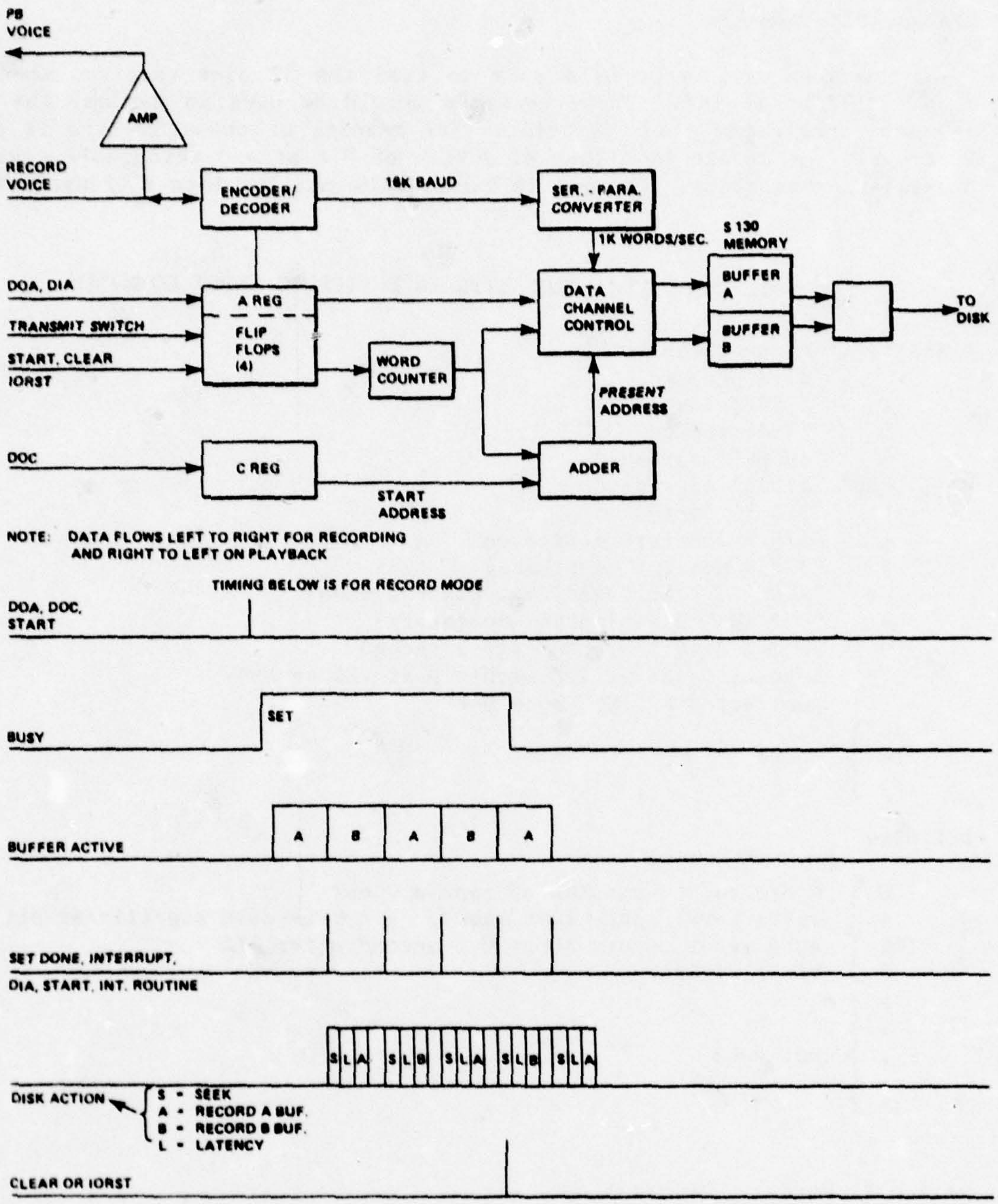


Figure 4. Device 31 Block Diagram and Timing

Device 31 Operation

In the description following, 256 word buffers are assumed. Longer buffers require more memory but save disk seek and latency time. Buffer length will be optimized and wired into the unit, however it will be changeable.

Recording

To place the unit in operation, a DOAS command should be issued, ensuring that bit 0 of the output word is 0 (for "record"). This sets the busy flip-flop, clears the done flip-flop, and enables the unit. The CPU also should issue a DOC command first, indicating the starting address of the first buffer to be used.

The unit starts filling the first buffer via the data channel. A buffer will fill in approximately 1/4 second. When 256 words have been stored, the done flip-flop is set causing an interrupt. Busy remains set. The interrupt service routine determines which buffer was filled, clears the done flip-flop and initiates a disk store operation of that buffer. In the meantime, the other buffer has been selected for filling with voice data. The second buffer will have a starting address 256 words greater than that of the first buffer. Approximately every 1/4 second this action will repeat, alternately filling the buffers.

There should be no need to change the contents of the C register which contains the start address of buffer A. If it is to be changed, it is recommended that it be changed by the interrupt routine.

A buffer-filling process will continue until the buffer fills following the clearing of busy.

The program must keep track of where on the disk the data are stored. Note that interrupts occur after the data are acquired. The time of acquisition as determined by the computer clock at interrupt time is, accordingly, the time of completion of the quarter-second block.

As long as busy remains set with the A register-bit 0 cleared, the process is repeated.

Playback

To playback the recorded voice, bit 0 of the device's A register must be set and the busy flip-flop must be set. The unit will then start reading out the contents of buffer A (which must previously have been filled from the disk). This starting buffer address is contained in the device's C register.

When buffer A has been read out, the done flip-flop will be set and an interrupt issued. Readout will continue from buffer B and then continue toggling between A and B every quarter-second setting done at the end of each buffer.

This action will continue until the busy flip-flop is cleared. When busy is cleared, read out will continue until completion of the current buffer. Done will be set for a final time, but no more buffer reads will be made.

Commands

Action of various commands and functions is shown in table 7.

TABLE 7. MEANING OF DEVICE 31 COMMANDS

Command	Function
Start	Sets busy and clears done. Busy must be active for the unit to record or playback. Start is used to initiate operation or to clear done when it is desired to continue operation. As the busy flip-flop is set, it causes buffer A to be selected as the buffer to be first used.
Clear	Clears busy and done. Device will continue its activity until the end of the current buffer in either the playback or record mode. It does not affect the A or C register.
IORST	Same as clear, but it also clears the A register (see below).
DOA	Sets/clears bit 0 of device A register: 0 = record mode 1 = playback mode
DIA	Reads bits 0 and 1 of device A register Bit 0 is as set by DOA Bit 1 is as follows: 0 = unit is selecting buffer A 1 = unit is selecting buffer B
DOC	Used to output starting address of buffer A. Buffer B starts 256 words higher. This command should not be issued when in the midst of a buffer operation. It can be changed at the completion of a buffer.

Effects on the System

While recording or playing back, the unit will require 1000 memory accesses per second. The associated disk action will require another 1000 accesses.

The unit will issue four interrupts per second and the disk will presumably issue eight interrupts per second (at the end of each seek and each sector read/write). The seek times will keep the disk fairly busy if it is assumed to be doing other transfers also. The maximum seek and latency time is 95 milliseconds. By using bigger buffers, this situation can be eased.

Disk capacity required is 2000 bytes per second of voice. On a five-minute exercise, assuming 25 percent voice time, we are recording 150,000 bytes or 1.5 percent of disk capacity for a 10 Mbyte unit. Continuous recording procedures would raise this requirement to 6 percent of disk capacity for a five-minute exercise.

The Trainee Panel

The use of this panel was discussed in the Functional Design Report.

This unit is part of the trainee station. Its location and size are as shown in Appendix A. The unit is designed by Logicon. It contains Logicon designed circuitry along with a speech preamplifier circuit from Threshold Technology and a Joystick digitizing circuit from Megatek. All trainee related controls and indicators are located on the front panel except for the foot switch.

Figure 5 is a view of the front panel. Table 8 describes each of the elements on the front panel.

The trainee panel is connected to the system controller by two 100-foot cables entering at the rear.

All power for the trainee panel is provided by two external DC power supplies through a single rear-mounted connector.

Electrical Design

Figure 6 is a simplified block diagram. In the upper left a serial stream of 32 bits and a separate clock are received. In about 30 microseconds these data are stepped into a shift register. The parallel output of the shift register determines the state of the lights and the audio alarm. Transients occurring during the shift register loading are too brief to interfere with the lights. One light is actually on the instructor panel and is controlled by a differential logic signal to that panel.

In the lower left is the transmission circuitry for panel switches, VGU active, trainee voice active and voice level. These data are parallel-loaded into a 32 bit shift register and stepped out in serial fashion to device 30 in CPU1. The data are also recirculated in the shift register.

Transmission is initiated by receipt of a BUSY level. It also occurs automatically whenever there is a change in any of the inputs to shift register section A. This is accomplished by a comparator which is interrogated 20 times per second to determine any difference in present switch positions from those previously transmitted and held in the shift register.

Other blocks to the right in the figure are self explanatory.

A detailed understanding of circuit operation may be obtained from the logic diagrams in Appendix C.

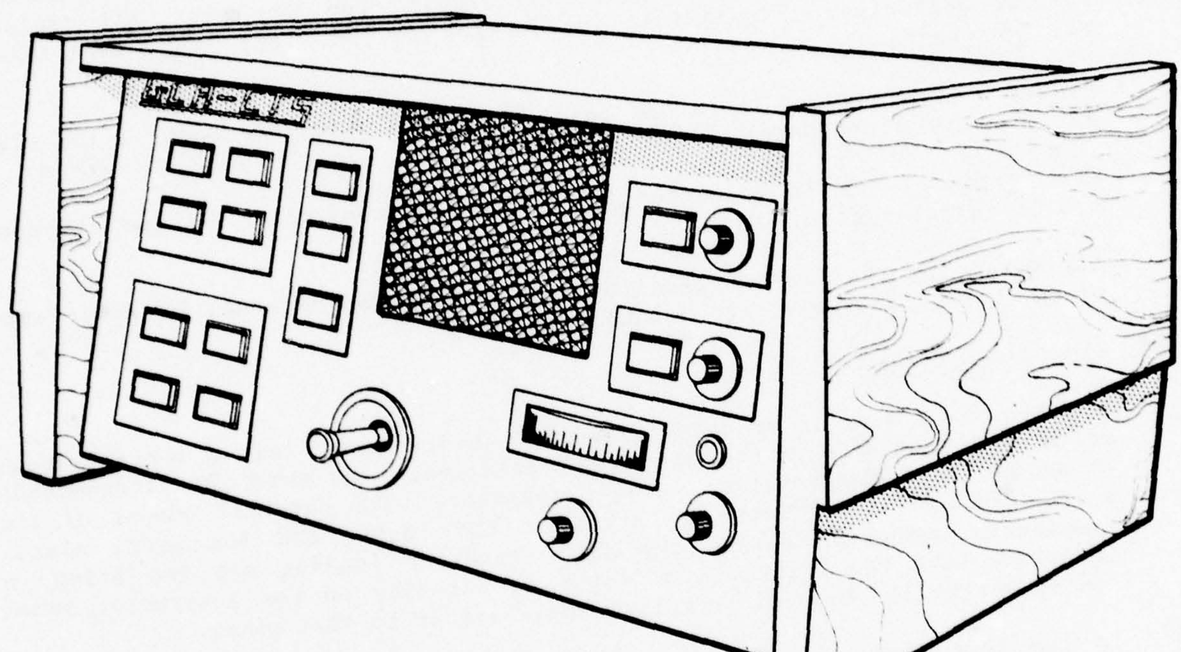


Figure 5. Trainee Panel, Front View

TABLE 8. TRAINEE PANEL INTERFACE ELEMENTS

Item	Description	Function
Servo Mechanism	A deflectable joystick extending 2" perpendicularly from panel Can be deflected 1-1/2" horizontally and/or vertically. Has spring return Has pushbutton in tip	Simulates antenna servo controls.
Radio Frequency Selection	A total of four square pushbuttons two for 270.8 megahertz, two labeled 318.8 megahertz. Buttons are alternate action. For each frequency one button is split with amber and green light table segments The other button has only an amber light.	The radio frequency panel consists of two sets of two button lights. In each set the first button light is the frequency select button. When the particular radio frequency is available for use the button light is not lit. When the frequency is in use the button is amber and the controller will hear an alarm in his/her headset if he/she selects the frequency. When the frequency is available and the controller selects it, the button light turns on and stays green until deselected. The second button light of the set is the monitor button light which the PAR controller selects when he/she wishes to monitor the communications between the pattern controller and the aircraft pilot. The amber light within the button comes on and stays on until the button is deselected.
Inter-Controller Communications System (ICS)	A total of four square alternate-action pushbuttons. Each contains a red light which may flash or be steady on or off. Labels are: 3 5 7 SUPER	The ICS is used by the PAR controller to communicate with the pattern controller or to monitor approaches conducted by other positions. The button light must be depressed in order for the PAR controller to communicate with the pattern controller. The button light will be illuminated as a red source when the button is depressed and will remain on until the button is deselected. 3 5 7 are for other controllers. SUPER is for the supervisor (instructor).

TABLE 8. **TRAINEE PANEL INTERFACE** ELEMENTS (CONT)

Item	Description	Function
Clearance System	One square alternate-action lighted pushbutton and two separate square lights. The switch light is white and is labeled "REQUEST." A green light is labeled "CLEARED." A flashing red light is labeled "W/O."	The GCA clearance light system consists of one button light and two individual lights. The system is used by the PAR controller to request landing clearance from the tower and by the tower to inform the controller of landing clearance or to cancel that clearance. The button light is illuminated as a white source when the button is depressed. The cleared light is a green light that indicates that the tower has granted the aircraft clearance to land. The second light is a red flashing light that indicates that the tower has cancelled the landing clearance. In addition to the flashing red light, an auditory alarm is also activated when the tower cancels clearance.
Volume Level Meter	A horizontal meter. Calibrated from 0 to 10. 0 - 4 Colored White 4 - 8 Colored Green 8 - 10 Colored Red	Shows trainee's voice level. Should be in green region when "FIVE" is spoken.
Volume Control	A knob with the numbers 1 through 5 on the skirt	Used to set the proper level on the Volume Level Meter
Audio Output Controls	Two alternate-action switches, lighted white when selected. One is labeled "HEADPHONE," the other is labeled "SPEAKER." Two knobs each having 300° of rotation. Each is labeled "LOUDNESS."	These allow either the headphones or the speaker to be energized at the level desired.
Speaker	A 4" dynamic loudspeaker	Audio output
Headset Jack	A 4-circuit jack for headset (microphone plus earphones).	Audio output

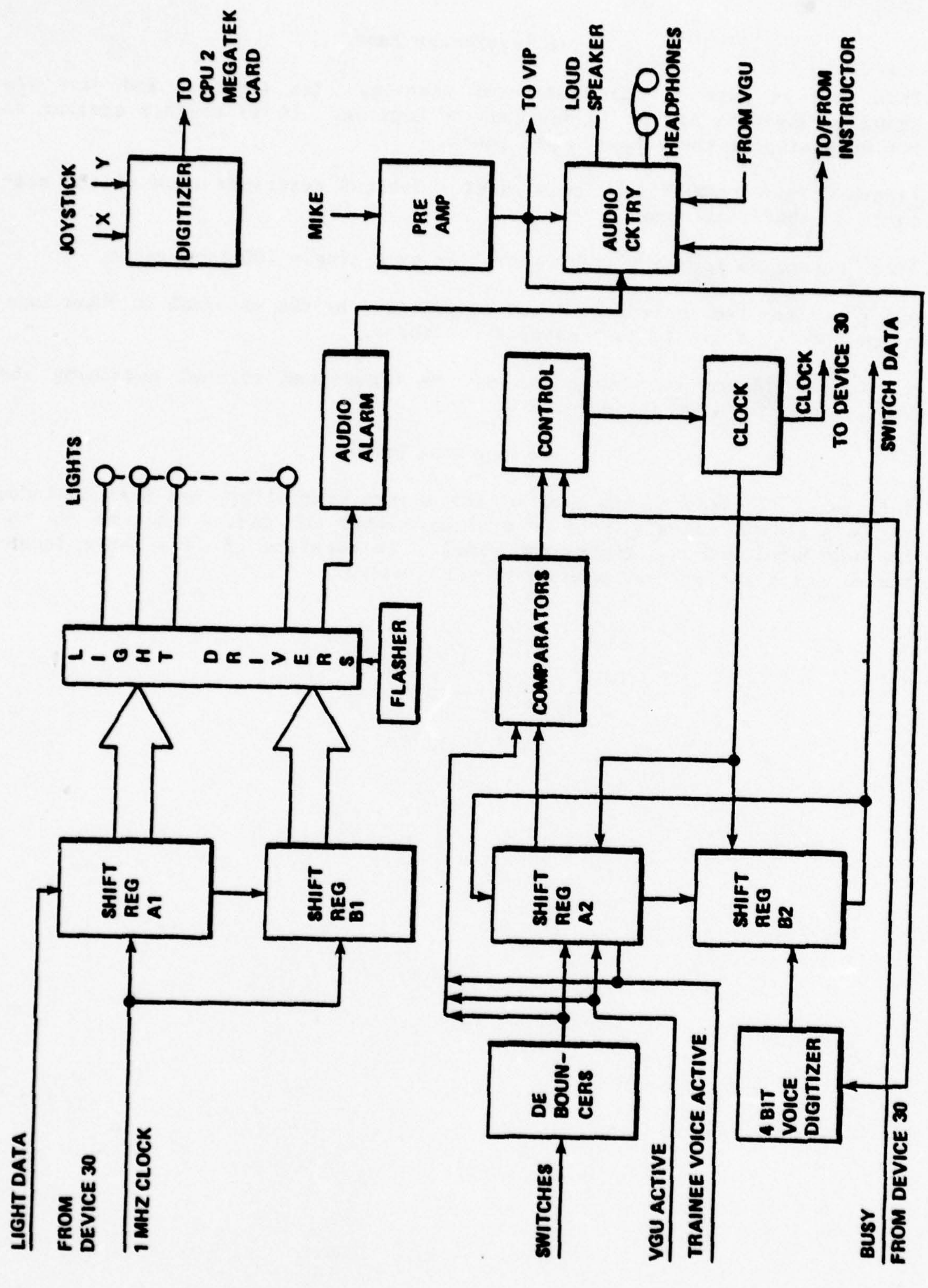


Figure 6. Trainee Panel Block Diagram

The Instructor Panel

This unit is part of the instructor station. Its location and size are shown in Appendix A. It is designed by Logicon. It is roughly similar to but much simpler than the trainee panel.

Figure 7 is a view of the front panel. Table 9 describes each of the elements on the front panel.

It is connected to the system controller by a single 100 foot cable.

All power for the instructor panel is provided by two external DC power supplies through a single rear-mounted connector.

Electrical design is simple and may be understood through examining the logic diagrams given in Appendix D.

The Junction Panel

This panel, located in the rear of the system controller, has been included to consolidate signals into a minimum number of cables running to the trainee panel and the instructor panel. It consists of five cable receptacles and a set of test points, properly wired.

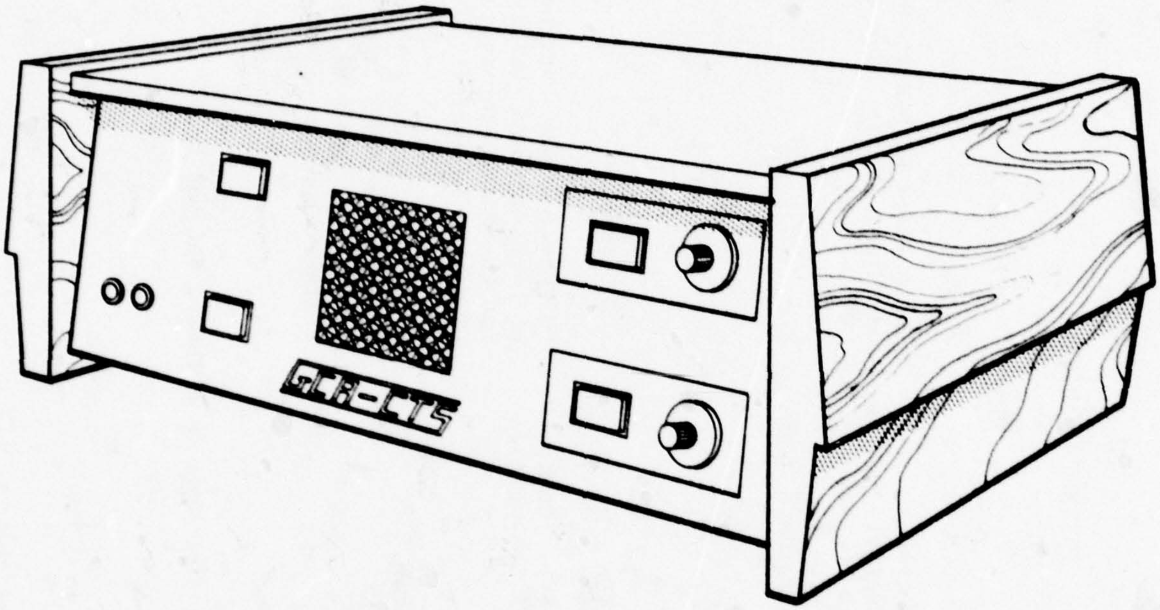


Figure 7. Instructor Panel, Front View

TABLE 9. INSTRUCTOR PANEL

Item	Description	Function
Audio Output	Two alternate-action switches lighted white when selected. One is labeled "HEADPHONE" and the other is labeled "SPEAKER." Two knobs each having 300° of rotation. Each is labeled "LOUDNESS."	These allow either the headphone or the speaker to be energized at the level desired.
Headset Jack	A dual jack to accommodate microphone and headphones	Audio output
Speaker	A 4" dynamic loudspeaker	Audio output
ICS	A square momentary switch, lighted. The light may be off, flashing red or steady red.	This button must have been depressed to signal the computer to turn the light on steady. It must be on for the instructor to talk with the trainee. (See note below.)
Audio Monitor	A square momentary switch, lighted	Repeatedly pressing the switch will toggle the light on and off. When light is on and trainee does not have SUPER depressed, the instructor can monitor VGU, trainee or device 31 playback.

Note: Unit can be in ICS or Audio Monitor mode or neither, but not both.

SECTION IV

THE SOFTWARE ENVIRONMENT

The software environment includes both the system support software used for normal operations and the diagnostics used for preventive maintenance and troubleshooting.

System Support

GCA-CTS will rely on vendor-supplied support software. Specifically, it will take advantage of the many features of the Real-Time Disk Operating System (RDOS) and will use the FORTRAN language. Most of the coding will be in FORTRAN 5, although some FORTRAN-compatible assembly language code will be needed. GCA-CTS will make use of a vendor-supplied graphics software package. The Data General RDOS, FORTRAN 5 and Macro Assembler are described briefly in the paragraphs that follow. A discussion of the Megatek graphics library routines is also included.

RDOS

RDOS was shown in the laboratory version to be capable of meeting the demanding real-time response requirements imposed by the GCA-CTS. It has the capability to schedule and allocate control to many different program tasks to provide simultaneous use of system resources and thereby maximize the efficiency of program operation.

The RDOS executive constitutes the main framework of the operating system, and it is resident in main memory at all times. Functions performed by this resident portion of RDOS include interrupt processing, overlay and buffer management, system call processing, and device interrupt servicing. Other modules of the system are brought into main memory from disk storage, as they are required to perform specific functions such as device initializations, file maintenance operations, and spooling control. In addition, the mapped RDOS used by GCA-CTS supports mapped memory addressing. The memory allocation and protection (MAP) unit provides a hardware separation of operating system areas from user address space. Moreover, it extends the maximum core configuration for a single CPU from 32K total to up to 32K for the resident operating system and up to 32K directly addressable by the user. (A foreground user also has up to 32K directly addressable memory, but this feature is not used in the GCA-CTS.) In a mapped system, two addressing modes exist. In the first mode, absolute mode, only the lower 32K is directly addressable and the mapping device is not used. RDOS resides in these low physical memory locations and executes in absolute mode.

The second mode is called mapped, or user mode. In user mode up to thirty-two 1024_{10} word blocks of memory are mapped by the management unit to produce an apparent (logical) 32K continuous address space. Any program operating in user mode uses a complete logical address space including its private page zero and extending through its upper memory bound. This upper bound is determined by the requirements of the individual program and it may extend as high as 32K. The operating system is responsible for assigning

free memory from its available pool to each user program prior to its execution. The technique used to manage the mapping unit and the construction of the user program in logical address space is also the responsibility of mapped RDOS.

Although mapped addressing extends the total amount of resident memory, it does not itself permit any single user program to exceed 32K words of memory. Since this restriction is unacceptable to some application programs, including GCA-CTS, RDOS provides two facilities for accessing the extended address space above 32K. Both virtual user overlays and window mapping create extended address space by storing data into memory blocks outside the 32K address space directly accessible by the user. When this program material is to be accessed, the desired blocks are remapped into the user's address space by enabling the memory management unit.

RDOS also provides the capability to bring in parts of a program from the disk as they are needed. The RDOS system can reserve portions of user address space for this function and divides it into fixed-length partitioned core storage areas which form a repository for programs of a limited size. This allows the RDOS user to segment a larger program into one or more parts which fit into the fixed-size core areas at execution time. These program segments are called user overlays and are stored on disk in core image format to facilitate rapid loading when their execution is required.

Other features of RDOS include full I/O support for a wide range of peripherals including the disk, CRTs, the printer, and the IPB.

An important function of any real-time operating system is the efficient handling of input-output operations. Optimum usage of matching devices and central processor time in the accomplishment of tasks is a major reason for designing and implementing a multitasking system. Since I/O devices are slow compared to the internal speed of the computer, they must be programmed to overlap their operations with computations, when possible, in order to increase usable CPU time by allowing one task to operate while I/O is in progress, to greatly increase efficiency of I/O operations, and to provide more throughput of data by removing bottlenecks caused by slow peripherals. The responsibility of RDOS I/O control is to react during normal program execution to the structuring of I/O requests, making assignments of requests to machine devices when they are idle, and queuing requests for devices which are busy. Through the queuing facility, RDOS makes it possible to achieve maximum and continuous overlap of many tasks without direct intervention by the tasks themselves.

The concept of a task is central to an understanding of GCA-CTS operation both at the level of I/O handling and at the applications program level. A task is a logically complete, asynchronous execution path through a program, subprogram, or overlay which demands use of system resources (usually CPU control). Many tasks may be directed to operate in a single re-entrant path, and each of these tasks may be assigned a unique priority. One real-time program may have from several to a virtually unlimited number of

logically distinct tasks. Each task performs a specified function asynchronously and in real-time. CPU control is allocated by the RDOS task scheduler to the highest priority task that is ready to perform or continue performing its function. This system scheduler and its associated routine together support the high level FORTRAN 5 tasking calls through the Universal Multitasking Interface (UMTI).

In addition to these runtime support functions, RDOS provides a powerful Command Line Interpreter program and also editors, compilers, assemblers, and debuggers which allow interactive software development to proceed in an efficient, user-oriented way.

FORTRAN 5

Compiler

The FORTRAN 5 compiler provides a mechanism for generating very efficient object programs from programs written in a superset of the FORTRAN language. Additional FORTRAN language features include:

- Full mixed mode numeric conversion.
- Acceptance of any expression as a control variable or parameter in a DO statement or DO-implied list.
- Generic library functions.
- Declarations that may appear anywhere in the program.
- No reserved words or reserved function names.
- All blanks ignored, except in Hollerith constants.
- IMPLICIT statement for data typing.

The efficiency of Eclipse FORTRAN 5 code derives from its full use of the powerful Eclipse instruction set and from the optimization of the generated code. Subscript computations, type conversion, comparisons, many library functions, etc., are generated in-line and can thus take full advantage of the compiler's interstatement optimization and ability to search for common expressions that need to be evaluated only once. Local optimization includes the following:

- Multiplication of an integer by a power of 2 is performed by shifting.
- Redundant operations, such as addition of the constant 0 or exponentiation to the constant power of 0 or 1, are eliminated.
- The compiler takes advantage of the associative or distributive properties of operators by reordering or eliminating some operations.

- Exponentiation of variables by positive integer constants is performed by in-line multiplication.

- Variable and expression values may be assigned to and remain in registers throughout a portion of the program.

- Floating point operations are optimized for effective use of floating point hardware. Hardware floating point operations will be performed asynchronously when possible.

Runtime Support

The FORTRAN 5 runtime library package supplies routines for performing integer, single and double precision real and complex mathematical operations, routines that perform formatted and unformatted input and output, and file creation and maintenance functions, routines that provide interfaces to system facilities, and routines that create and maintain a multitask environment and provide overlay management facilities. All routines in the library are re-entrant, permitting one or more tasks to enter and execute a routine before prior executions are complete. This means that many tasks can share a single copy of a routine, and significant core savings result. GCA-CTS will also make use of the Load-On-Call Overlay Facility (LOCO) which automatically loads and releases overlays as required.

Runtime Environment

Figure 8 illustrates the runtime configuration of main memory for a multitask FORTRAN 5 program. RDOS resides in a separate address space and so is not shown. Three levels of data must be distinguished in this environment:

- Per-ground data are common to the entire ground; that is, data which are global to all tasks (if more than one task exists) but which are local to a given ground (if foreground and background both exist in a mapped system). They include all linkage to runtime routines in page zero, certain information maintained by RDOS about the ground, common blocks and static storage, the runtime file table, and all executable code.

- Per-task data are particular to a given task. They include task status and priority; the values of the accumulators, carry, and program counter; the state of the floating point unit; the FORTRAN 5 state variables; I/O control information; and the task's runtime stack.

- Per-routine data are particular to a single execution of a runtime routine. They are stored in a distinct portion of a task's runtime stack known as a frame. Using different stack frames for different activations of the same routine makes the routines re-entrant, as required for multitask operation.

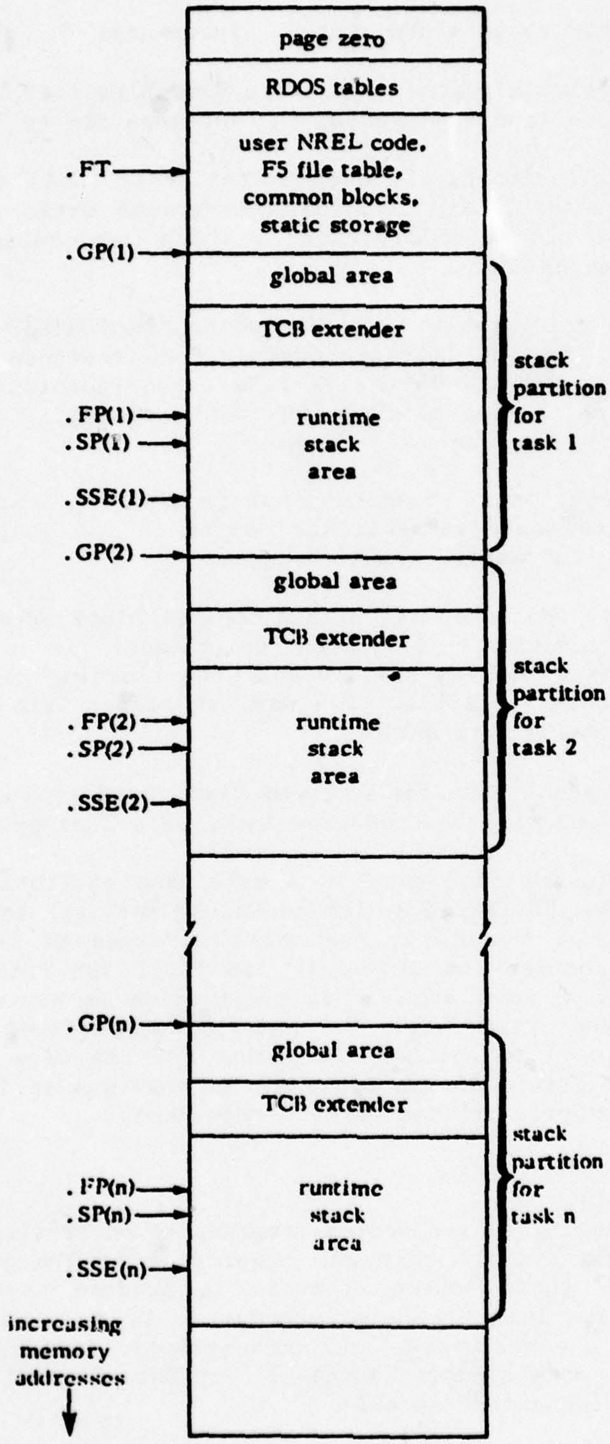


Figure 8. Runtime Memory Allocation

Figure 8 shows how these data are stored. Shown are:

1. Page zero, containing the task state variables (.FP,.SP,.SSE,.RP, and .GP); and locations reserved by the hardware and by RDOS.

2. RDOS tables, including the User Status Table (UST) and an (optional) overlay directory, which contain per-ground data; and the pool of task control blocks (TCB), each of which may contain per-task data for a single task.

3. User normal relocatable code (including the FORTRAN 5 main program, subprograms, runtime library routines, and routines from the system library), the FORTRAN 5 runtime file table (pointed to by page zero location .FT), common blocks, and static storage. All data in this area are per-ground data.

4. Per-task global area, containing mainly I/O task control information. The page zero state variable .GP points to the start of the global area for the currently executing task.

5. For multitask environments, a task control block extender, containing additional per-task information which must be maintained for a FORTRAN 5 task, such as the state of the floating point unit and the FORTRAN 5 state variables. The word at offset TELN of a task's TCB points to its TCB extender.

6. The runtime stack area for a given task, used by compiled FORTRAN 5 programs and runtime routines for local data storage.

As shown in the figure, each task in a multitask environment has its own per-task global area, TCB extender (where values for its state variables are stored when it is not executing; just as the values of its accumulators, carry and program counter are stored in its TCB), and runtime stack area. By default, the memory available at initialization is divided equally into as many stacks as there are TCBs. Fortunately, a partition macro is available which can be used to specify the number and the size of stack partitions. This is of critical importance in an environment like the GCA-CTS where many tasks compete for limited core resources.

Macro Assembler

In general, an assembler allows source programs to be written using familiar characters to create symbols that are meaningful to the programmer. The assembler processes these source programs to produce object programs in machine language, meaningful to the computer. To do this, the assembler simply substitutes a numeric code for each symbolic instruction code and a numeric address for each symbolic address. The Data General macro assembler includes the following added features:

1. Expanded expression evaluation that provides for explicit as well as implicit precedence. The class of operators includes relational operators.

2. A powerful macro facility which allows complete recursion as well as nested macro calls.

3. An assembly repeat feature for producing many lines of source from a simple repeat construct. This facility also encompasses conditional assembly. Conditionals may be nested to any depth.

4. An assembly suppression feature that allows the programmer to suppress assembly until a given label is encountered.

5. The assembler can generate a three-digit number to replace a symbol anywhere in assembly code. Thus, the digits may be part of a symbol or number or may stand alone. The feature is useful, for example, in providing unique labels during table generation.

6. A class of special symbols having a value (like ".") related to an internal assembler variable. This class of symbols allows the user to determine useful information such as the number of arguments specified by a current macro call. Further, many pseudo-ops have a value associated with them and, using the proper syntax, may be used within expressions.

7. Literal references by any memory reference instruction. All literals will be optimally resolved in page zero. Literals are not restricted to absolute numeric quantities and, in fact, may consist of any legitimate expression.

8. The assembler has the facility to generate three-character alphanumerics for each occurrence of the character $ within a label. The facility is implemented in such a way that, for example, unique labels can be generated within nested macros.

Megatek

The Megatek display processor series 5000 provides the support for the graphics portion of the GCA-CTS. The Megatek consists of a processor and a large cathode ray tube (CRT). It contains no user memory, that being provided by the Eclipse and accessed through a DMA cycle-stealing device. Software provided with this system was designed for an unmapped machine and the routines were not compatible with FORTRAN 5. Therefore, it was necessary to modify the software to perform in a mapped FORTRAN 5 environment.

The Megatek features vector graphics, full translation in the X- and Y-planes, and a hardware generated character set. Pictures are created within a display list with a series of microprocessor instruction codes. These codes inform the Megatek processor of the desired location of the CRT beam, and whether the vector to be drawn is visible or blanked. It is possible to append, insert, delete, or write over any picture component within the list. The screen can be referenced in screen units (4096 x 4096) or units defined by the user. Pictures can be modified dynamically, changing and replacing pictures very rapidly.

For optimal use of the Megatek processor, a Megatek graphics package is being used. This package contains a series of FORTRAN callable subroutines which build the display list, thus making Megatek usage extremely simple. There are routines to draw lines, move the CRT beam without drawing lines, translate pictures, including rotation, and enlargement or shrinkage of pictures. There are also routines to draw the hardware generated characters, and activate the joystick. All of these access the display list, either directly, or through a lower level assembly language routine. Other programs provide support for the main routine. These include programs to convert floating point numbers to ASCII characters, to change the limits of a picture, or of the joystick, and to provide the coordinates of the joystick position.

Diagnostics

A wide range of diagnostics will be supplied with the GCA-CTS. These will be used for routine preventive maintenance, to ensure system integrity after shipping and installation, and for trouble-shooting hardware failures. Diagnostics provided by vendors and by Logicon are described in the following pages.

Data General Corporations's Diagnostic Operating System

DDOS is an operating system that has been developed by Data General Corporation to provide an efficient and systematic method of running diagnostic tests on DGC processors and peripheral equipment.

DDOS may be used most efficiently for problem isolation and detection by following simple procedures.

A special debugger program has been included with DDOS, which allows the operator to isolate sections of a diagnostic by the setting up of breakpoints.

DDOS is available in either cartridge disk, diskette, or magnetic tape form.

The minimum equipment requirements for using DDOS are a DGC processor, a terminal, and a magnetic tape, diskette or disk drive. DDOS contained on magnetic tape will run successfully in processors having a minimum of 4K words of memory. DDOS contained on diskette will run in processors having a minimum of 8K words of memory. Some diagnostics require more space than the minimums specified above. If a diagnostic should be scheduled that is too large for the memory in the processor, an error message will be printed on the terminal to so inform the operator. For reference purposes, it is necessary to have a listing of each diagnostic program that will be run.

The operator controls device testing with simple online commands issued to the processor through a terminal keyboard. These are interpreted by the DDOS monitor to allow the operator to load and run any diagnostic contained on the tape or disk. In addition to specifying diagnostics, commands are used to determine which I/O devices will be tested. Certain commands also allow the operator to run a data channel test concurrently with a diagnostic.

DDOS constructs a table in memory, called the equipment table, containing the mnemonic and device code for each piece of equipment on the processor data bus. Devices contained in this table will be tested automatically when certain commands are issued. Devices which are not included in the equipment table automatically may be added to it with simple DDOS commands. DDOS will only support DGC diagnostic tests. Any user written diagnostics will have to run separately from this operating system.

Table 10 gives a list of the Data General diagnostics which are applicable to the GCA-CTS hardware.

Most of the DGC diagnostics consist of a series of simple tests, each of which sends a particular combination of input signals to a small portion of the unit under test, and performs some simple test on the output. Generally, each of these tests is initialized by a small subroutine. This subroutine sets the internal pass counters, that is, the number of passes to be made through the test, establishes the proper address for the diagnostic to jump to after each pass, and determines any other parameters necessary to run the test. Another small subroutine keeps track of the number of times the diagnostic has been run successfully. This subroutine is responsible for having the diagnostic jump to a particular address after it has been run the number of times that had been established with the internal pass counter. If the diagnostic has not completed the established number of runs, this small subroutine forces the diagnostic to begin again.

Simple Diagnostics

On some of the simple diagnostics, a failing test will simply cause the processor to halt. The accumulators will contain some information about the failure and the address lights will give the location of the failure. The user must consult the listing of the diagnostic test that was running to find the exact reason for the failure. A failure in this type of diagnostic almost precludes passing any other diagnostic.

Complex Diagnostics

On the more complex diagnostics the stop or halt instruction will be imbedded in a subroutine. This subroutine will force the diagnostic to print the memory location where the failure occurred and return control to the monitor program. If a command has been issued which causes all applicable diagnostics to run in sequence continually, the monitor will print the name of the diagnostic in which the failure occurred in addition to printing the location of the failure, each run through the entire list, excluding the first.

TABLE 10. DATA GENERAL DIAGNOSTICS

Title	Function
ECLIPSEA: Central processor diagnostic part 1	Tests arithmetic and logical operations.
ECLIPSEB: Central processor diagnostic part 2	Tests bit manipulation instructions, accumulator compare and logical shift instructions.
ECLIPSEC: Central processor diagnostic part 3	Tests logic of memory reference instructions, auto-increment and decrement, etc.
ECLIPSED: Central processor diagnostic part 4	Tests stack manipulations, extended operations, etc.
ECLIPSEE: Central processor diagnostic part 5	Tests logic of two-word instructions, etc.
ESPCLEX: Special exerciser	Test all instructions, mapped and unmapped, with ERCC option.
ECLIPSE**: Exerciser parts 1-9	Tests reliability of CPU instructions.
EIMRT S: Multi-program reliability - short	Tests CPU, memory, floating point, map and character instructions.
EIMRT L: Multi-program reliability - long	Like EIMRT S, but also tests primary disk and printer.
EIMRT P: Multi-program reliability - peripherals	Like EIMRT S but also exercises peripherals.
EMMPUA, EMMPUB: Memory allocation and protection unit test	Tests the MAP feature.
ECLSC: Semiconductor memory test	Tests semiconductor memory.
EXMEM: Extended memory exerciser	Tests memory, taking MAP and interleaved memory into account.
EMLER: ERCC diagnostic multilayer CPU2	Error checking and correction test.
EPFAIL: Power shutdown test	Tests power monitor and auto restart option.
IPBR: Inter-processor bus reliability.	Tests the various types of IPB transfers.

TABLE 10. DATA GENERAL DIAGNOSTICS (CONT)

Title	Function
IPBD: Inter-processor bus diagnostic	Tests a single IPB board.
EIOA, EIOB: I/O tests	Verifies operation of the I/O features, I/O bus, interrupt and data channel and VCT instruction.
40DI: 4010/4023 or 4077/4078 diagnostic	Tests specific I/O boards.
ETTY: Teletype test	Detects malfunctions in the teletype logic.
LCD: Video display test	Checks 6052, 6053 video displays.
RTCTST: Real-time clock test	Real-time clock maintenance.
CDF: Cartridge/diskette formatter	Formats disks and diskettes.
CDR: Cartridge/diskette reliability	Exercises disk controller and drives.
EIFUPX: Floating point firmware execiser	Tests floating point instruction set reliability.
EIFPUD: Floating point firmware diagnostic	Detects failures in floating point unit.

Overnight Testing

DDOS was designed such that control is returned to the monitor if an error occurs in a complex diagnostic so that the system will not hang up on a single failure when a relatively long run is attempted, such as overnight. When control returns to the monitor, the next scheduled test, if any, will be loaded and run. Consequently, to determine whether any errors have occurred, the terminal or line printer output must be examined.

Program Modes

Diagnostics are executed under DDOS in one of four modes: auto, semi-auto, manual, or debug. The primary differences between the modes lie in the associated operator communication for each diagnostic.

Auto Mode. In auto mode DDOS compares its equipment table against the equipment requirements of the test programs as shown in the directory, and sequentially executes those programs that exercise the devices on the machine under test. Each test program is loaded and executed automatically. At its conclusion the test program returns control to DDOS so that the next program can be run. No operator communication is required after the initial command.

Semi-Auto Mode. If communication with the operator is required, (to establish what surfaces of multi-surface disk are to be tested by a disk reliability program, for example) the program may be run in semi-auto mode. Programs to be run in semi-auto mode will not be run in auto mode; however, in all other respects operation is similar. One or more test programs may be specified in the initial command to DDOS and these programs will be executed sequentially with return being made to DDOS after each program ends.

Manual Mode. Manual mode is used when a return to DDOS is not desired. The initial command results in DDOS loading the program and starting it; however, at the conclusion of the test, the test program loops back to the start of the test. This mode is useful when the operator is troubleshooting a machine failing a particular test. The LOAD and CLOAD commands cause DDOS to operate in manual mode.

Debugger Mode. If the DEBUG command has been given, the debugger is loaded at location $30{,}000_8$ for the Eclipse line processors, then the test program is loaded. DDOS transfers control to the debugger rather than to the starting address of the test program. This procedure is useful for things other than debugging. For example, a program to test the disk drive can be loaded with DEBUG and the program will not run until the operator starts it. This gives him time to dismount the DDOS disk and mount a scratch disk.

Once the test program is started from the debugger, operation is identical to manual mode. The test program will loop on completion without returning to DDOS. It should be noted that the test program is free to write over the debugger, and if this occurs, the debugger becomes useless after the program starts. Individual diagnostic listings should be consulted to determine whether or not the debugger will be overwitten.

Megatek Diagnostic Program

A vendor-supplied diagnostic program exercises the graphics display processor. It provides a variety of test patterns which can be used to observe picture alignment, refresh rate changes, blinking, and the various levels of intensity. Hardware faults can be traced by observing their effect on the test patterns. These test patterns are also used in display alignment procedures. The diagnostic is supplied as a stand-alone program which is BOOTed in from the disk or diskette.

Tally Diagnostic

The Tally printer has a self-test mode of operation which is selected by a switch setting on the side of the unit. The test exercises every character in every position. Detection of functional failures requires only a cursory glance at the one page test pattern.

Logicon Diagnostics

TTI 500/Votrax

A diagnostic program will be provided to ascertain both the Votrax and the Threshold 500 are in working condition. The procedure consists of:

1. Placing the headset onto the Votrax speaker in the prescribed manner.
2. Setting all Votrax and headset input preamplifier controls to the requested positions.
3. Running a Threshold 500 test program, NEWDIAG, in conjunction with a Votrax test program, COUNT.

The Threshold 500 listens while the Votrax counts from zero through nine. This is repeated three times during a run. The input feature count for each of the 32 features is kept and compared to a benchmark range. A failure is indicated for each feature count discrepancy. A file, VSPDATA, holds all test results. If voice recognition errors occur consistently, this diagnostic program may pinpoint the malfunction.

High Speed Correlator

A high speed correlator (HSC) resides on the Threshold Technology supplied 6050 interface board. The HSC is utilized to perform the correlation between voice reference patterns and input feature patterns. Two diagnostics are available for this device.

HSC2DIAG compares HSC outputs to software calculated correlation sums. Inputs to this test program are random numbers. A random number seed is requested upon entry by the program. The test terminates after 20 errors or 65,536 comparisons.

HSCDIAG compares HSC outputs to correlation sums obtained by using a bit counter device which resides on the 6050 board itself. The bit counter responds to DIC and DOC of the Eclipse instruction set when addressed via the Threshold 500 device code. This test compares all possible integer pairs but will abort after 20 errors. A complete test run without errors requires several days. However, half an hour of runtime should be sufficient. If the 6050 board has failed completely, this test will not be able to discern errors.

Both diagnostics store their results in a disk file, HSCERR. It is recommended that both diagnostics be run to ensure dependable HSC operation.

Trainee Panel Diagnostic

The trainee panel diagnostic may be used to affirm panel functionality or to determine the source of malfunctions. The tests check the performance of both the interface card within CPU1 and the trainee panel device.

A "turnaround" test is utilized to check out the interface card. Bit patterns are sent to the interface card via DOA, DOB (data output commands). Assisted by a special test fixture the bit patterns are read back in (via DIA, DIB commands) and compared with the original output. If the two are equal, interface card function with respect to the CPU is assured. Several sets of bit patterns are shuttled to thoroughly test all data lines.

Part two of the diagnostic consists of an interactive test which requires a human monitor. The program requests various trainee panel key inputs and acknowledges their reception via the CRT. Notification is given of the intent to activate alarms and lights. After activation the computer awaits confirmation via keyboard input. The Votrax and the digitized audio device also participate in this phase of diagnosis. After Votrax output, a one second timeout is noted on the CRT. A voice input is requested and the system waits for one second of silence before the recorded voice is played back. The peak input level is displayed on the CRT. The record-playback sequence may be repeated until satisfaction is achieved.

A log of discrepancies for the entire diagnostic run is provided for output.

SECTION V

SOFTWARE DESIGN

Overview

The Training/Functional Design Report described a system which would both provide instructional materials and an environment for practicing GCA control skills. From the trainee's perspective, the GCA-CTS has five major modes of operation:

- Phase 1
- Phase 2
- Phase 3 and P-run
- Replay
- Demonstration

Reviewing briefly, phase 1 provides multimedia presentations and demonstrations to teach the various topics in the syllabus and to elicit speech samples for vocabulary reference pattern creation.

Phase 2 is an optional freeze and feedback mode in which the student practices and the system freezes if a mistake is made on the new material. The mistake is explained and the student is given the opportunity to try again.

Phase 3 provides a simulated environment in which the trainee can practice the newly acquired skills and integrate them with old skills. This simulated environment is at first simplified somewhat, but as the student progresses through the syllabus it becomes more realistic. The P-run, or performance test, is the student's final examination. It is just like other phase 3 problems with the exception that special scoring options are available.

Several replay options are available for every phase 3 problem, thereby enabling the student to review his performance.

Finally, in the demonstration mode the system conducts approaches utilizing a simulated final controller. This mode is utilized for instructional purposes in phase 1, and it also operates whenever the system is otherwise idle. This provides a natural transition to alignment checking procedures when the student signs on.

These modes of operation are woven together in the GCA-CTS as shown in figure 9. As this figure indicates, the initialization routines start keyboard processing and IPB I/O routines, and initiate the demonstration mode. When a student signs on to the system, the training control program takes over and selects one of the instructional phases based upon the course

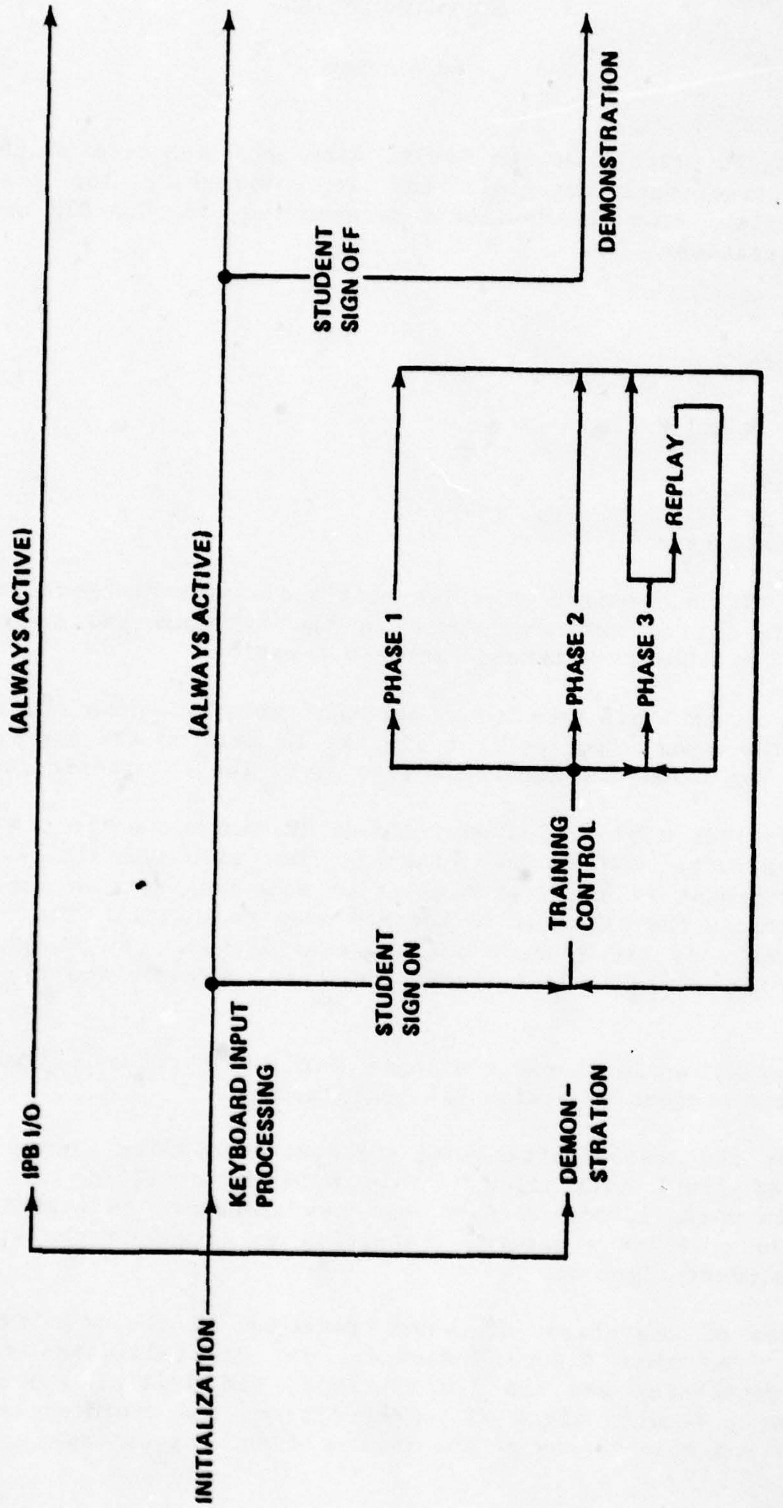


Figure 9. Overview of GCA-CTS Processing

syllabus and the student's progress to date. Each time the student completes a phase of instruction, the training executive regains control and selects the next mode. When the student signs off the system, the demonstration mode is initiated again.

The figure also indicates that the keyboard task remains active at all times, which allows the various special requests to affect the course of training.

The IPB I/O task is likewise shown as always active, which reveals another layer of complexity to this simple scheme. The actual processing burden is divided between two computers which communicate via the full duplex lines of the IPB. These two processors, dubbed CPU1 and CPU2 in previous sections, can be thought of as the training controller and the speech recognition and display processor respectively. CPU1 is the master and is responsible for controlling the modes of operation described above. CPU2 is devoted primarily to processing speech input data from the Threshold 500 for voice data collection and speech recognition, and to display processing. It also accepts keyboard entries from the trainee console and maintains IPB communications with CPU1. Task allocation by CPU is transparent to the user with the exception that the startup protocol requires that the master computer be started first. From then on the GCA-CTS is one system, providing a range of training capabilities from computer-aided instruction through a final examination in a realistic radar environment.

Design Philosophy

The design goal of a flexible instructional system was deemed best met by the development of table driven executives. A single executive is responsible for each phase of instruction and uses ASCII text files to provide the required variety of experiences. Changes to the course of instruction are implemented by simply editing the text files and do not require recompilation or reloading of the GCA-CTS executable routines.

The master file is the syllabus file which is used by the training control program. This syllabus contains an ordered list of file names and an indicator of the phase of instruction to which each corresponds. There is one of these (indicator, file name) entries for each phase of every task in the syllabus. Training control initiates the specified phase executive which then uses the information in the file to provide training or practice situations. These files are described in Appendix F.

Looking more closely at the requirements of the various modes of operation, it becomes apparent that they share many of the same functional elements. Table 11 shows these functional elements and the modes of operation to which they apply. The GCA-CTS routines which will satisfy these requirements have therefore been designed to be general enough to operate as required in the various modes.

The design has been approached in a top-down fashion. The discussion in the remainder of this report is organized according to a functional hierarchy which regards training control with its phase executives as primary, the major functions as secondary, and so on.

TABLE 11. FUNCTIONAL ELEMENTS OF THE GCA-CTS MODES OF OPERATION

Applicable Functions	Demonstration	Mode Phase 1	2	3 P-run	Replay
Voice data collection		X			
Speech recognition			X	X	
Speech understanding			X	X	
Aircraft, pilot, environment	X	X	X	X	
Radar	X	X	X	X	
Display	X	X	X	X	X
Model controllers	X	X	X	X	
Performance measurement			X		X
Keyboard input processing	X	X	X	X	X
IPB I/O processing	X	X	X	X	X
Trainee panel input processing		X	X	X	
Trainee panel output processing	X	X	X	X	X
Votrax output processing	X	X	X	X	X
Speech digitizer input processing		X		X	
Speech digitizer output processing	X	X	X	X	X
User clocks	X	X	X	X	X

Communication between routines takes place by means of variables in labeled common and through disk files. These data structures are defined in Appendixes E and F, respectively. The common variable and parameter definitions given therein include the intertask communication message keys and event numbers.

Program Descriptions

The design of the GCA-CTS software is detailed in the subsections which follow. For each topic a narrative overview is provided with block diagrams where appropriate. This is followed in most cases by a program description and string chart for each routine. The program description gives the purpose of each routine and indicates its relationship to the GCA-CTS environment by specifying how it is activated, which routines it calls or activates (exclusive of WAKEUPs and XMTs), the common variables used, and the files referenced. The language is specified as F (FORTRAN 5) or A (assembly). The string charts outline the processing within the routine. Appendix G describes string chart format rules.

Training Control, Modes of Operation

This section is concerned with the description of training control processing. It begins with a discussion of overall training control; then the executives which handle each of the modes of operation are described.

Training Control

A training executive is responsible for selecting the instruction the student will encounter. This includes the selection of regular problems as well as remedial exercises. The GCA-CTS syllabus is designed to draw the student through a carefully constructed, step-by-step skill acquisition process. Thus, every student completes every training experience. No exercise is skipped because each lesson contains material that must be mastered. This does not mean that the system is not adaptive, however. The student's performance is used to determine the number of practice sessions he must complete for each task. Furthermore, if the system detects that performance on selected previously acquired skills is declining apparently due to the addition of new material, it will simplify the problems presented. Finally, once a new skill is mastered, the need for remedial training on previously acquired skills is assessed, and such training is automatically initiated.

The system is also adaptive to the student's expressed needs. He can request a repeat of any training exercise he has previously completed. He can also elect not to use the freeze and feedback mode (phase 2) for any problem.

The routines described in the following pages include the training control executive, the problem selection module, and the module which adapts problem difficulty to the student's needs.

TZEC

Description: The training system executive initiates individual phase executives as required.

Entry point: TZEC

Classification: Task

Period: None

Language: F

Activated/called by: SGNON, PHAZ1, PZ2, PZ3A, PZ3B, PZDEMO

Cancelled by: TZEC

Activates/calls: IPBOUT1, SMENU, IMENU, SGNOFF, PHAZ1, PHAZ2, PHAZ3, TIMOUT, PZREQ

Cancels: TIMOUT

Input arguments: None

Common variables: MNST, MNIN, CL50, CLTGH, CLTICK, NCSYL, NCSCR, NCSR, GZSOFG, TZCSYL, TZCSR, GZCHAL, GZPHZ

Files created/changed: SCRATCH

Files referenced: SYLLABUS, student records

PROCEDURE

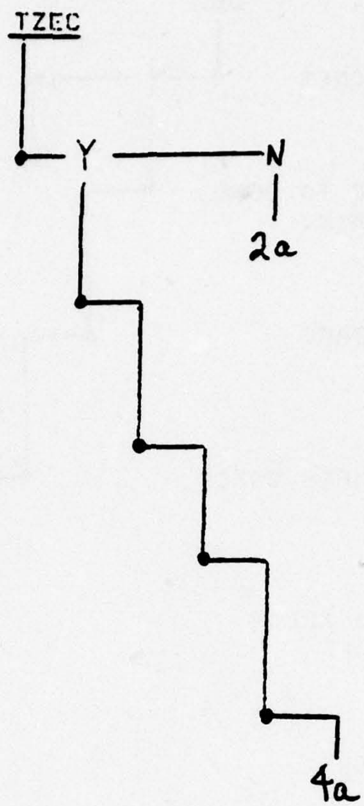
DID DEMO START TZEC
(PHASE=6?)

SELECT AN APPROPRIATE
ALIGNMENT PROBLEM

NOTIFY DISPLAY

START ALIGNMENT CHECK
SCORING (P19)

WAIT EVPHZ



TZEC FLOWCHART (SHEET 1 OF 5)

IS IT TIME TO GO HOME?

SET TIME TO GO HOME TO 2500
SO WE DON'T ASK AGAIN.

SELECT "HOME?" MESSAGE
NUMBER

SET GZSOFG SO SIGNOFFS COME
HERE

SET STUDENT MENU TO ALLOW
ONLY PROCEED, SIGNOFF

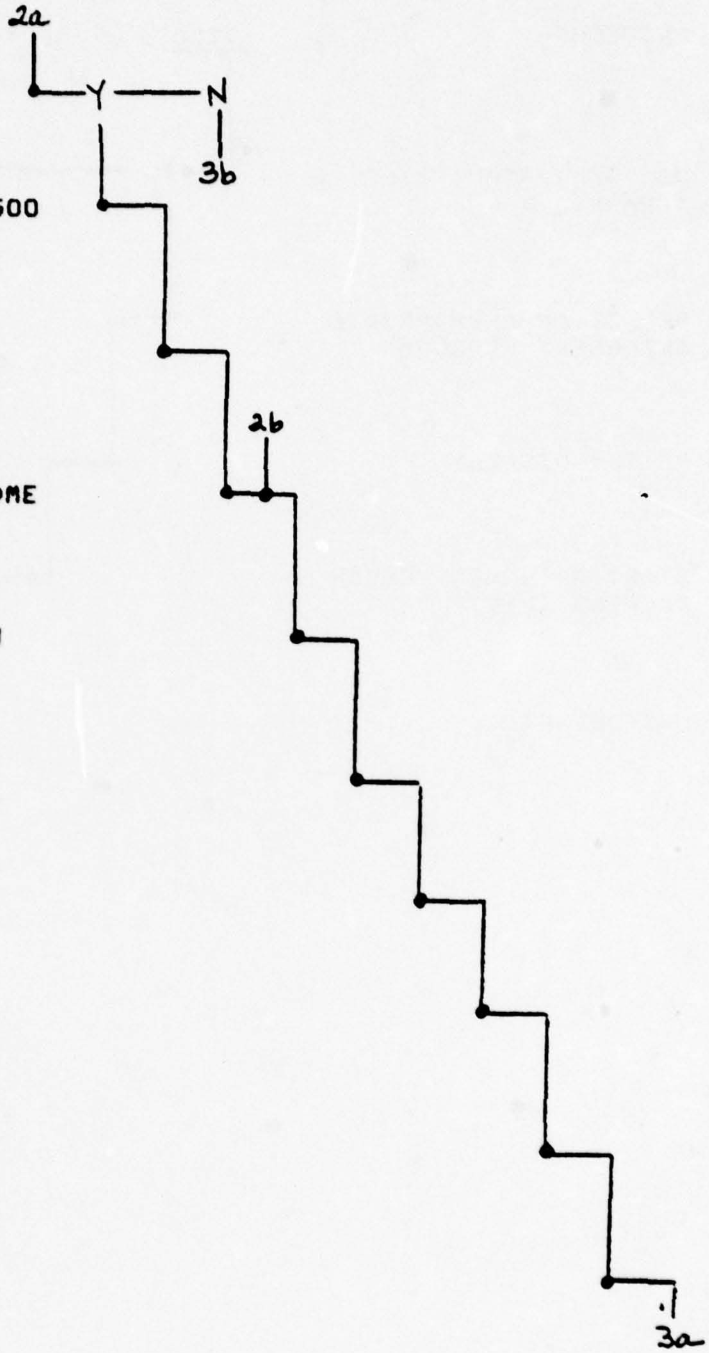
SEND QUERY TO STUDENT

START TIME OUT TASK

WAIT FOR MESSAGE FROM
STUDENT, OR TIME OUT

MESSAGE RECEIVED

RESET GZSOFG



TZEC FLOWCHART (SHEET 2 OF 5)

WAS THIS A TIME OUT MESSAGE?

INDICATE TIME OUT IN STUDENT RECORDS

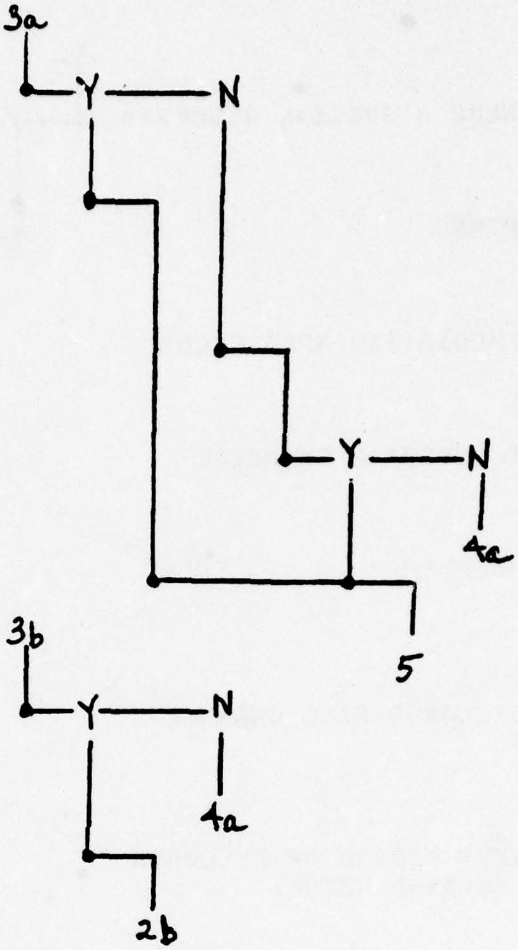
KILL TIME OUT TASK

WAS THIS A SIGN OFF MESSAGE?

CALL SGNOFF

HAS 50 MINUTES OF TRAINING ELAPSED SINCE BREAK?

SELECT "BREAK?" MESSAGE NUMBER



TZEC FLOWCHART (SHEET 3 OF 5)

WAS THERE A SPECIAL REQUEST?

CALL PZREQ

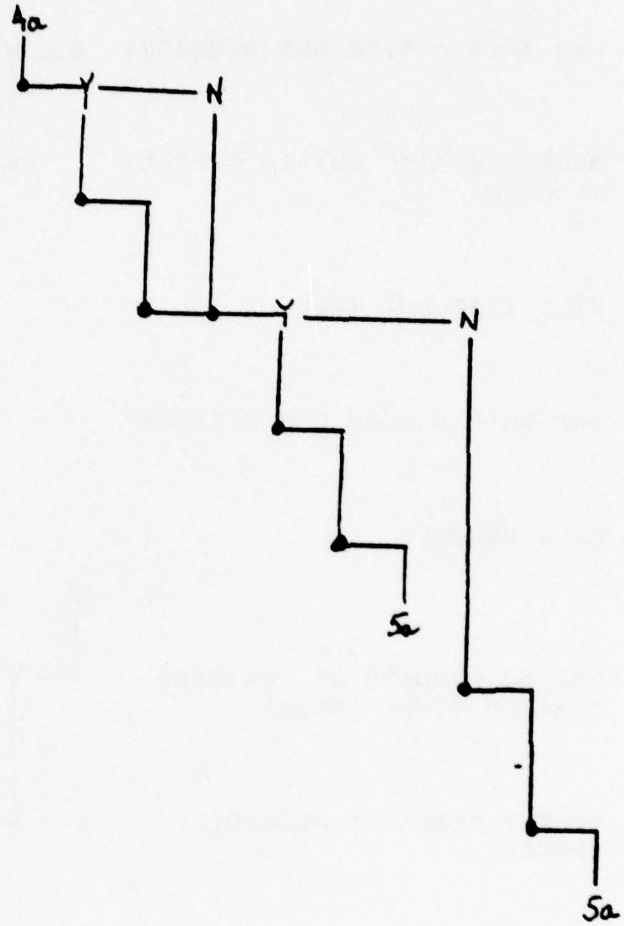
WAS REMEDIATION SPECIFIED?

CHOOSE REMEDIAL EXERCISE

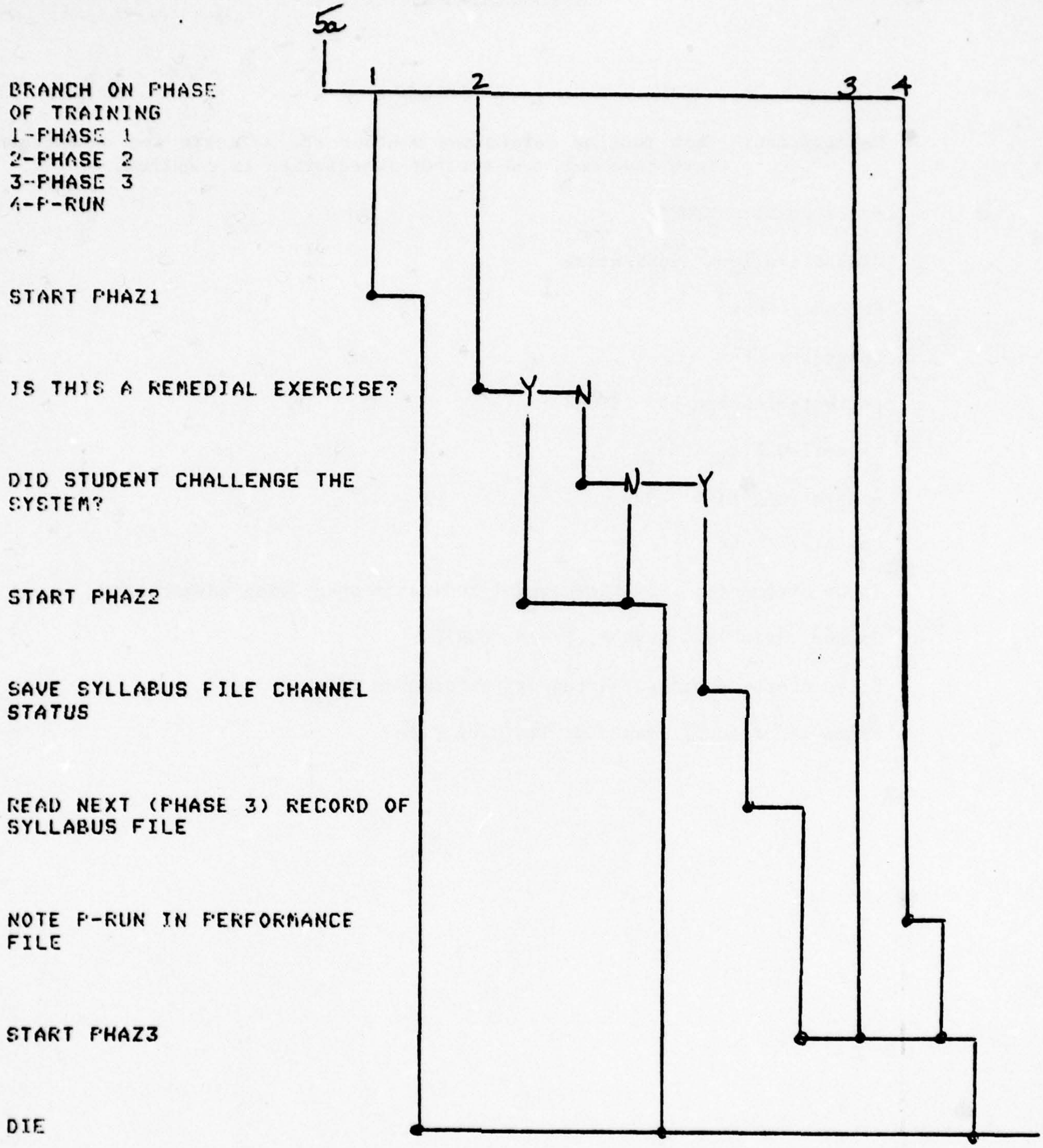
SET PHASE TO 7

SAVE SYLLABUS FILE CHANNEL
STATUS

READ NEXT RECORD OF SYLLABUS
FILE (SETTING GZPHZ)



TZEC FLOWCHART (SHEET 4 OF 5)



TZEC FLOWCHART (SHEET 5 OF 5)

SELECT

Description: This routine determines whether the criteria for advancement have been met, and whether remediation is required.

Entry point: SELECT

Classification: Subroutine

Period: None

Language: F

Activated/called by: SCORE

Cancelled by: N/A

Activates/calls: None

Cancels: None

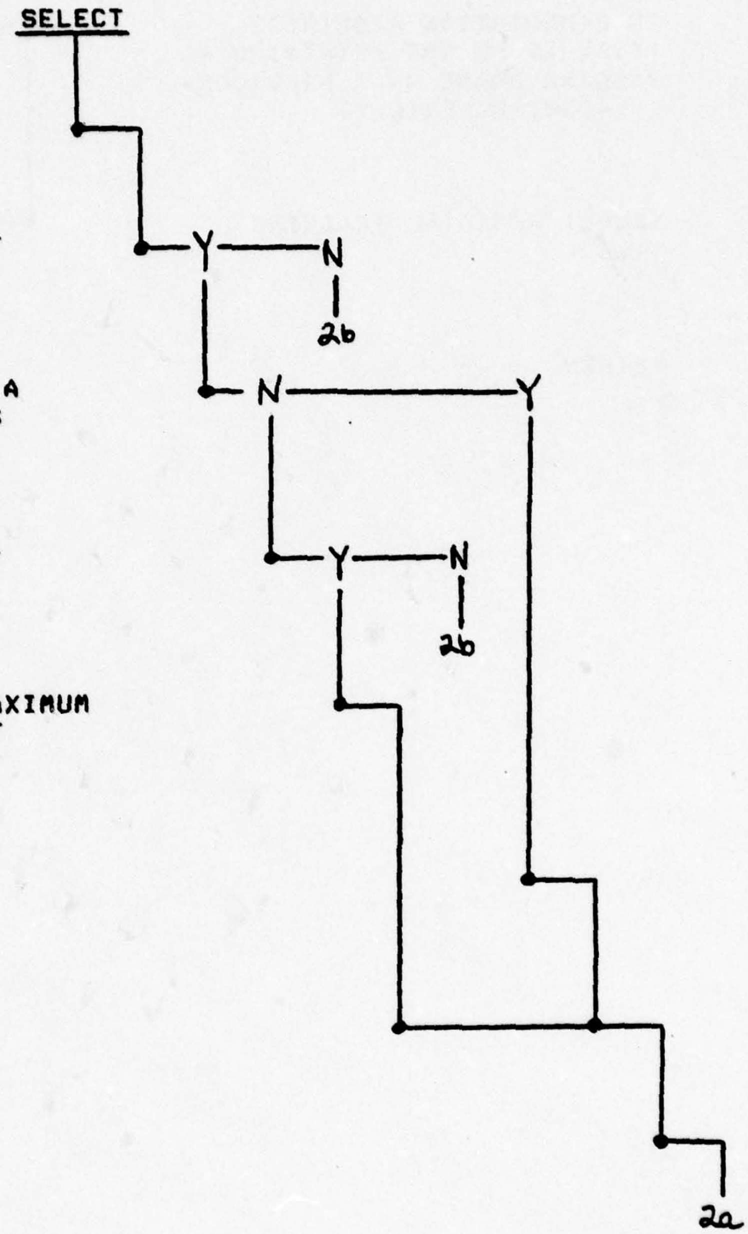
Input arguments: ADVANCE-Logical indicator specifying advancement

Common variables: PVN**, PVE**, GZPILL

Files created/changed: Student performance file

Files referenced: Remedial training file

PROCEDURE



SET ADVANCE FALSE

HAS THE MINIMUM NUMBER OF RUNS BEEN COMPLETED?

IS THE STUDENT ATTAINING A PASSING GRADE ON ALL PMVS RELATING TO THIS SKILL?

HAS THE MAXIMUM NUMBER OF RUNS BEEN COMPLETED?

NOTE IN STUDENT FILE: MAXIMUM RUNS COMPLETED, SKILL NOT MASTERED

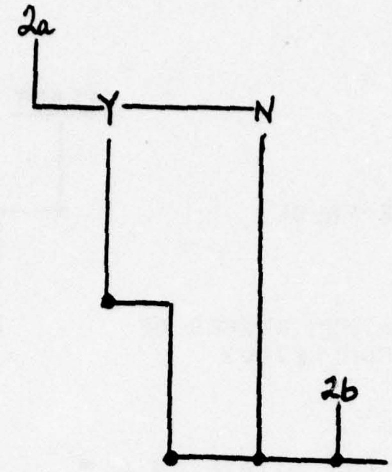
NOTE IN STUDENT FILE: THIS SKILL MASTERED

SET ADVANCE TRUE

COMPUTE TASK PERFORMANCE SUMMARY (TFB)

SELECT FLOWCHART (SHEET 1 OF 2)

IS REMEDIATION REQUIRED?
(I.E. IS HE NOT ATTAINING A
PASSING GRADE ON A PREVIOUS-
LY ACQUIRED SKILL?)



SELECT REMEDIAL TRAINING
TASK

RETURN

SELECT FLOWCHART (SHEET 2 OF 2)

ADAPT

Description: This routine is called by the phase 3 control routines to reduce the difficulty of a problem based upon the student's performance. Only selected PMVs are examined. Reduction in problem difficulty is accomplished by providing a slower aircraft, steady wind, and/or a better pilot. When these changes are made, they are noted in the student file.

Entry point: ADAPT

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ3A, PZ3B

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: ACTYP, ACSPD, ENFLUCT, PTYP, PTFLT, PVN**, PVE**

Files created/changed: Student performance file

Files referenced: None

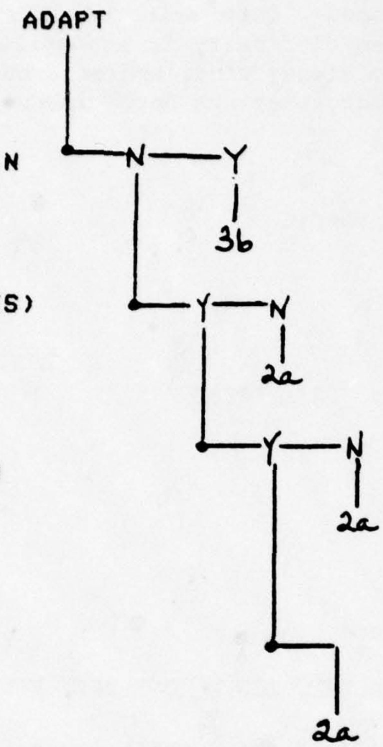
PROCEDURE

IS THIS THE FIRST PROBLEM IN THIS PHASE?

WAS PV05 (HEADING ADVISORIES) SCORED ON LAST LESSON?

IS THE SCORE FOR THE LAST RUN SIGNIFICANTLY LESS THAN THE AVERAGE SCORE ATTAINED ON THE LAST LESSON?

SET WIND FLUCTUATION TO 0



ADAPT FLOWCHART (SHEET 1 OF 3)

WAS PV06 (AZIMUTH POSITION AND TREND) SCORED ON LAST LESSON?

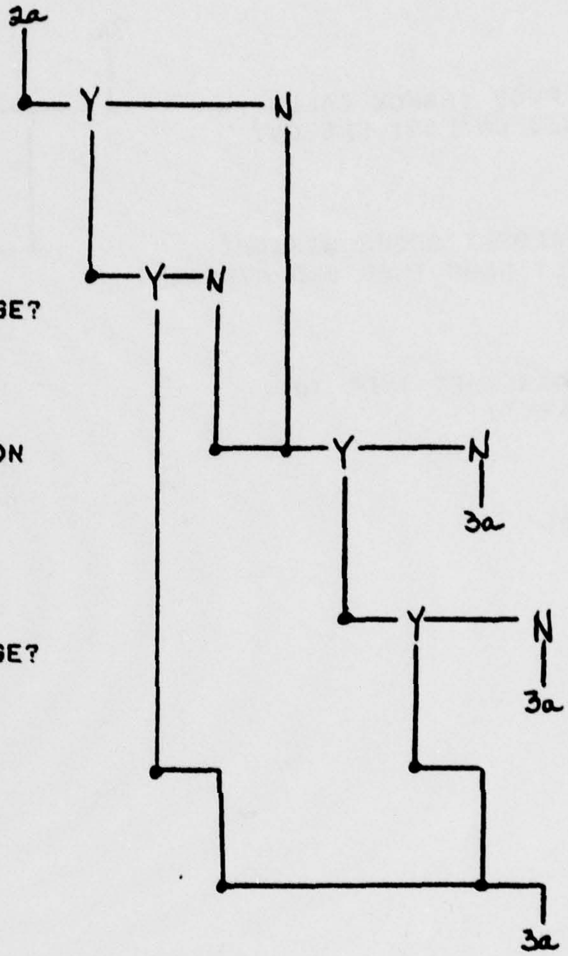
IS CURRENT SCORE SIGNIFICANTLY LESS THAN OLD AVERAGE?

WAS PV07 (GLIDEPATH POSITION AND TREND) SCORED ON LAST LESSON?

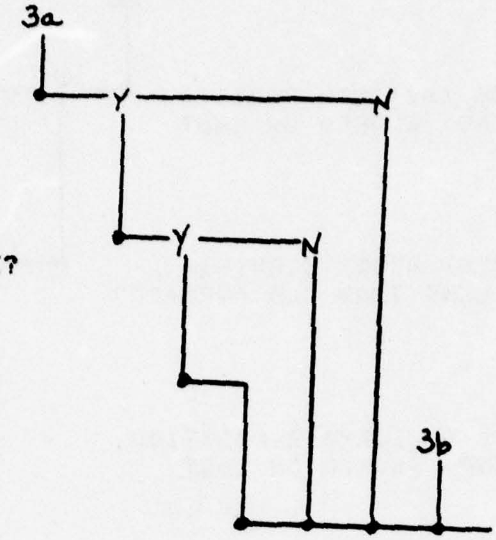
IS CURRENT SCORE SIGNIFICANTLY LESS THAN OLD AVERAGE?

SET PILOT TYPE TO 1 (BEST)

SET AIRCRAFT TYPE TO 1 (SLOWEST)



ADAPT FLOWCHART (SHEET 2 OF 3)



WAS PV08 (RANGE CALLS)
SCORED ON LAST LESSON?

IS PRESENT SCORE SIGNIFI-
CANTLY LESS THAN OLD AVERAGE?

SET AIRCRAFT TYPE TO 1
(SLOWEST)

RETURN

ADAPT FLOWCHART (SHEET 3 OF 3)

Phase 1 Training Executive

PHAZ1, the phase 1 training executive, handles the processing of phase 1 task files. Phase 1 task commands initiate voice data collection, Megatek displays, terminal CRT, and digitized voice prompts, waits on student responses, aircraft demonstrations, servo conditions, and task file sequencing logic. Explanations of command types and their associated instruction formats are included in the discussion on phase 1 task file structure in Appendix F.

Voice data collection is requested through subroutine P1VDC. P1VDC simply sends arguments to the SPEECH task on CPU2 via the IPB.

Display instructions are also handled in a similar manner by P1DIS. The task IMAGES is the receiver on CPU2 in this case.

Prompts, under the jurisdiction of P1PRM, are directed to the appropriate output device controller. The student CRT prompts are transferred via the IPB. The model controller prompt option also relies on the expertise of the EXPERT modules to select proper model controller advisories.

Radar servo conditions are initiated and frozen by P1RAD. Azimuth and elevation servos are positioned, aligned, activated, and deactivated via the IPB messages to SERVO on CPU2. Servo alignment conditions are also initialized by P1RAD via a common block for the RADAR routine.

Aircraft approach simulation initial values are set by P1AC. The simulation is begun on request, at which time a task which freezes on an event parameter is activated.

A couple of wait conditions, handled by P1WAI, are also applicable to aircraft simulation events, such as, wait for aircraft to enter azimuth zone. Other types of general wait conditions are also provided.

Task file sequencing instructions provide skips, if statement constructs, and subroutine constructs. Abnormal subroutine returns also provide returns to an offset from the normal return point. Nested subroutines are permitted up to five levels.

All instructions are defined in card format to form a primitive interpretive training language. Any phase 1 task file may be easily modified by replacing, inserting, or removing cards with care given to preserving sequence instruction validity.

Refer to figure 10 for an overview of phase 1 processing.

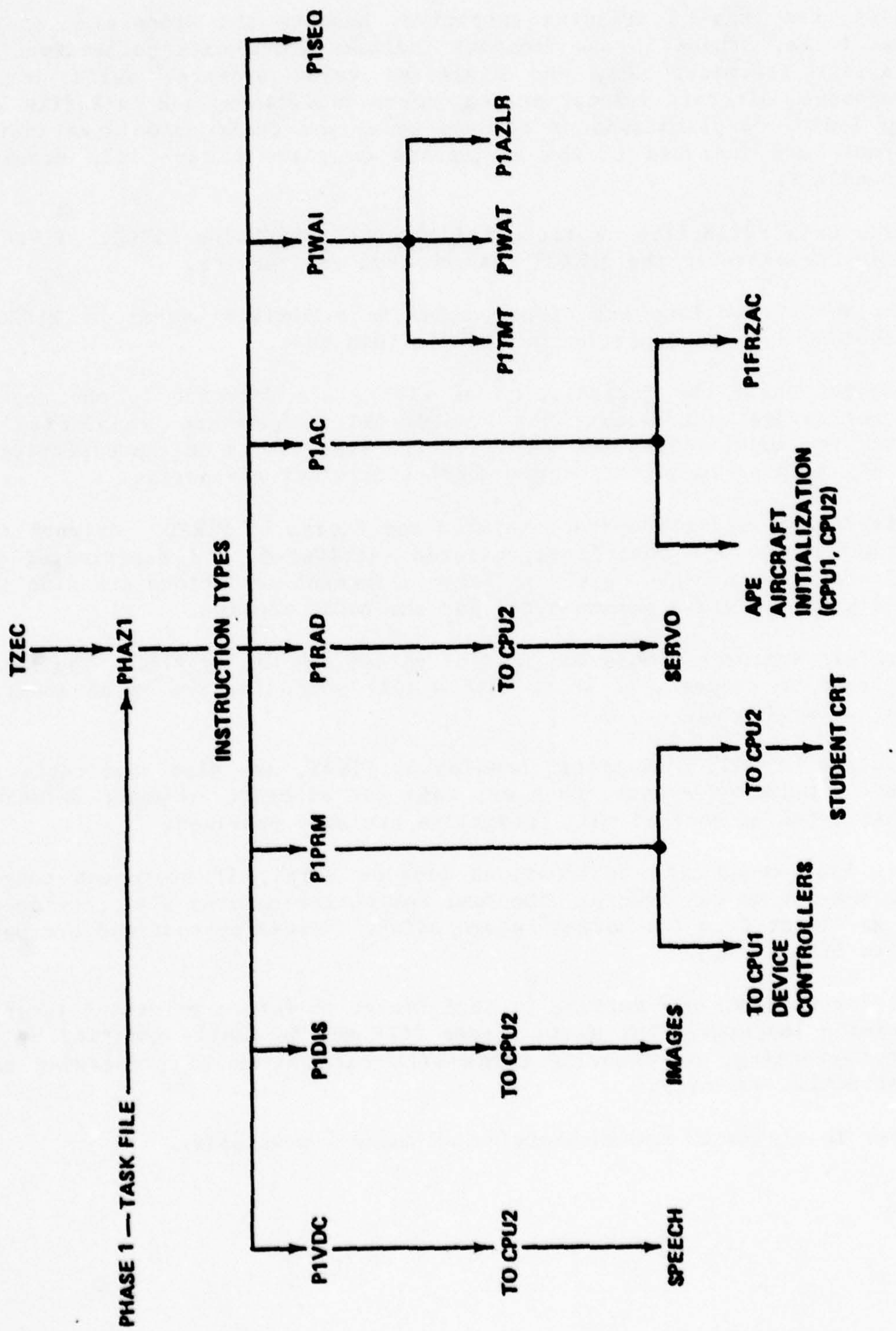


Figure 10. Phase I Processing Hierarchy

PHAZ1

Description: PHAZ1 is the training executive for phase 1 instruction during which the proper use of radio terminology is taught while formulating student voice reference patterns.

Entry point: PHAZ1

Classification: Task

Period: None

Language: F

Activated/called by: TZEC

Cancelled by: Self

Activates/calls: P1VDC, P1DIS, P1PRM, P1RAD, P1AC, P1WAI, P1SEQ, TZEC

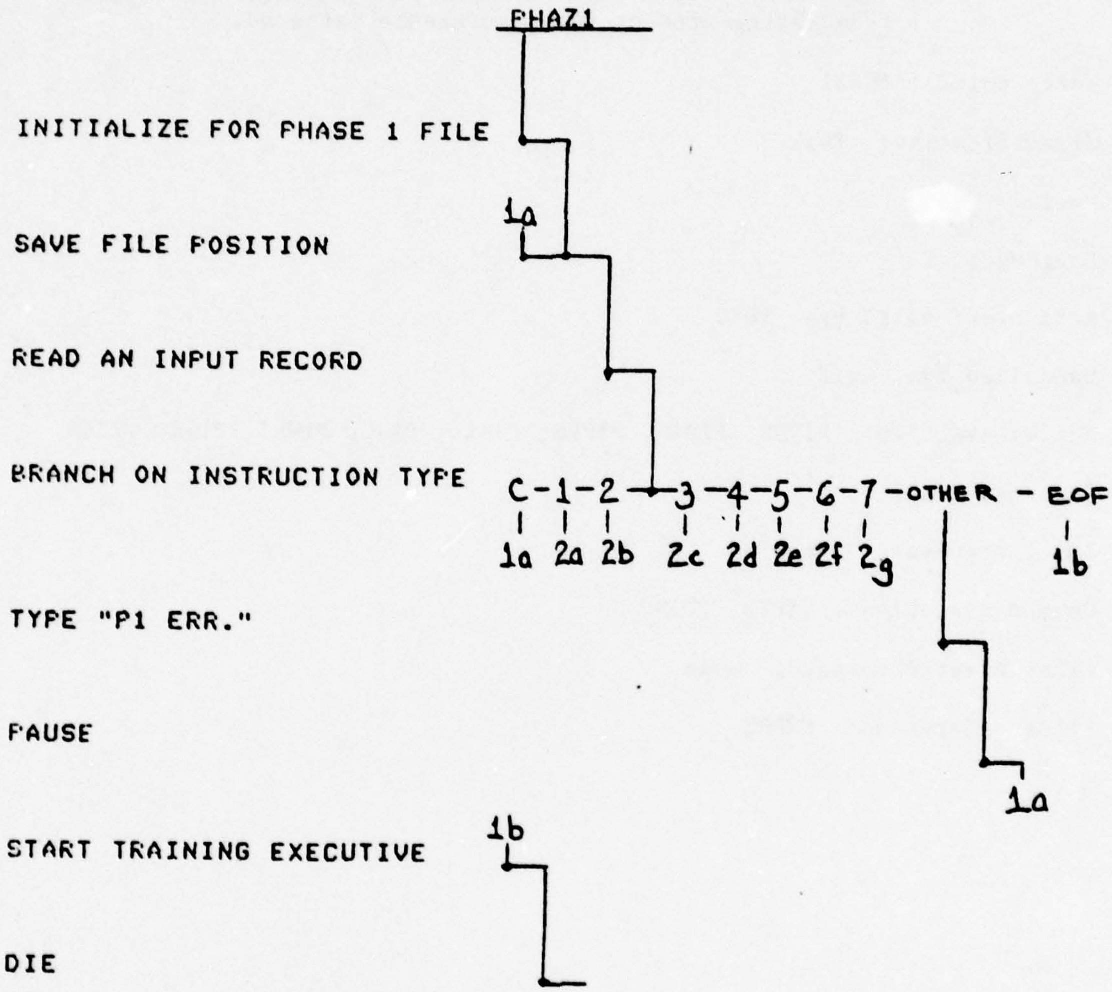
Cancels: Self

Input arguments: None

Common variables: FZPTR, FZINP

Files created/changed: None

Files referenced: FNPHZ



PHAZ1 FLOWCHART (SHEET 1 OF 2)

INITIATE VDC	2a	1a
INITIATE DISPLAY	2b	1a
INITIATE PROMPTS	2c	1a
INITIATE RADAR SERVO	2d	1a
INITIATE A/C SIMULATION	2e	1a
INITIATE A WAIT CONDITION	2f	1a
INITIATE A SEQUENCE CHECK	2g	1a

PHAZ1 FLOWCHART (SHEET 2 OF 2)

P1VDC

Description: Initiates phase 1 voice data collection.

Entry point: P1VDC

Classification: subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: IPBOUT1

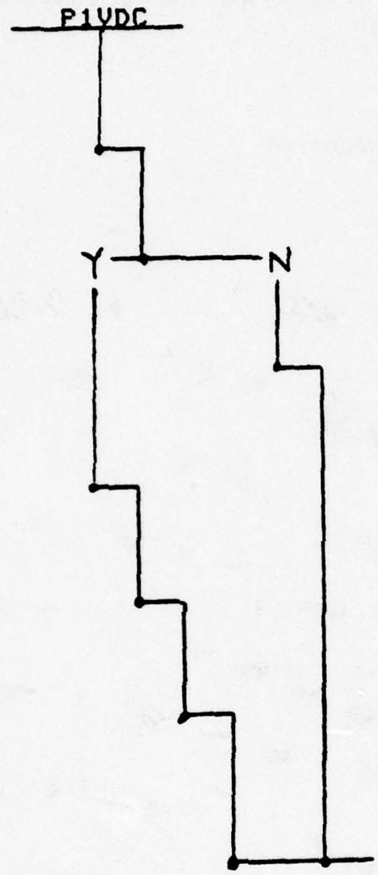
Cancels: None

Input arguments: None

Common variables: FZINP, PRDEV

Files created/changed: None

Files referenced: None



READ ARGUMENTS

IS IDENTIFIER VALID?

TYPE "P1 VDC ERR."

FIND PROMPT DEVICE

EXECUTE VDC INSTRUCTION

WAIT UNTIL VDC
INSTRUCTION COMPLETED

RETURN

P1VDC FLOWCHART (SHEET 1 OF 1)

P1DIS

Description: Phase 1 routine which initiates requested displays on the Megatek

Entry point: P1DIS

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: IPBOUT1

Cancels: None

Input arguments: None

Common variables: FZINP

Files created/changed: None

Files referenced: None

READ DISPLAY ARGUMENTS

BRANCH ON IDENTIFIER

TYPE "P1 DIS ERR."

SEND PICTURE ON MESSAGE TO
IPB (CALL IBPOUT1)

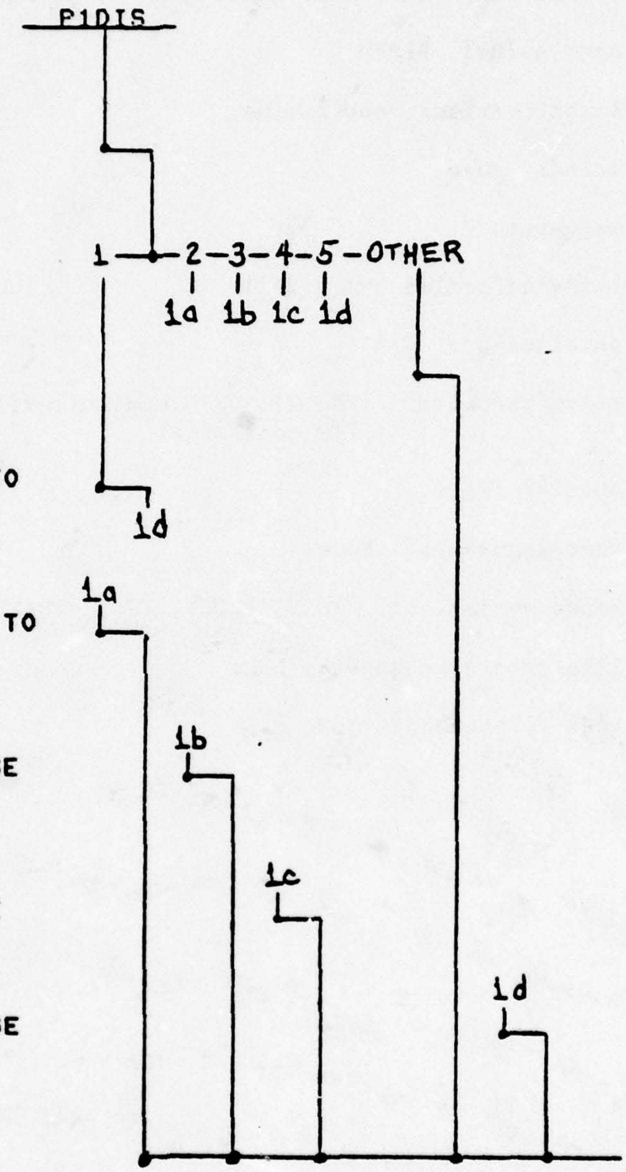
SEND PICTURE OFF MESSAGE TO
IPB (CALL IBPOUT1)

SEND START DISPLAY MESSAGE
TO IPB (CALL IBPOUT1)

SEND STOP DISPLAY MESSAGE

SEND BLINK PICTURE MESSAGE

RETURN



P1DIS FLOWCHART (SHEET 1 OF 1)

P1PRM

Description: Initiates generation of prompts for phase 1.

Entry point: P1PRM

Classification: subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: IPBOUT1, $VRO controller, Digitized audio controller, CPU1 $TTO controller

Cancels: None

Input arguments: None

Common variables: CTDEV, PRDEV, CTON, PRATL, FZINP

Files created/changed: None

Files referenced: None

READ ARGUMENTS

BRANCH ON PROMPT ID

TYPE "P1 PRM ERR."

SEND STUDENT CRT OUTPUT
MESSAGE (VIA IBPOUT1)

INITIATE VOTRAX MESSAGE
OUTPUT (CALL VSCON)

INITIATE DIGITIZED AUDIO
MESSAGE OUTPUT

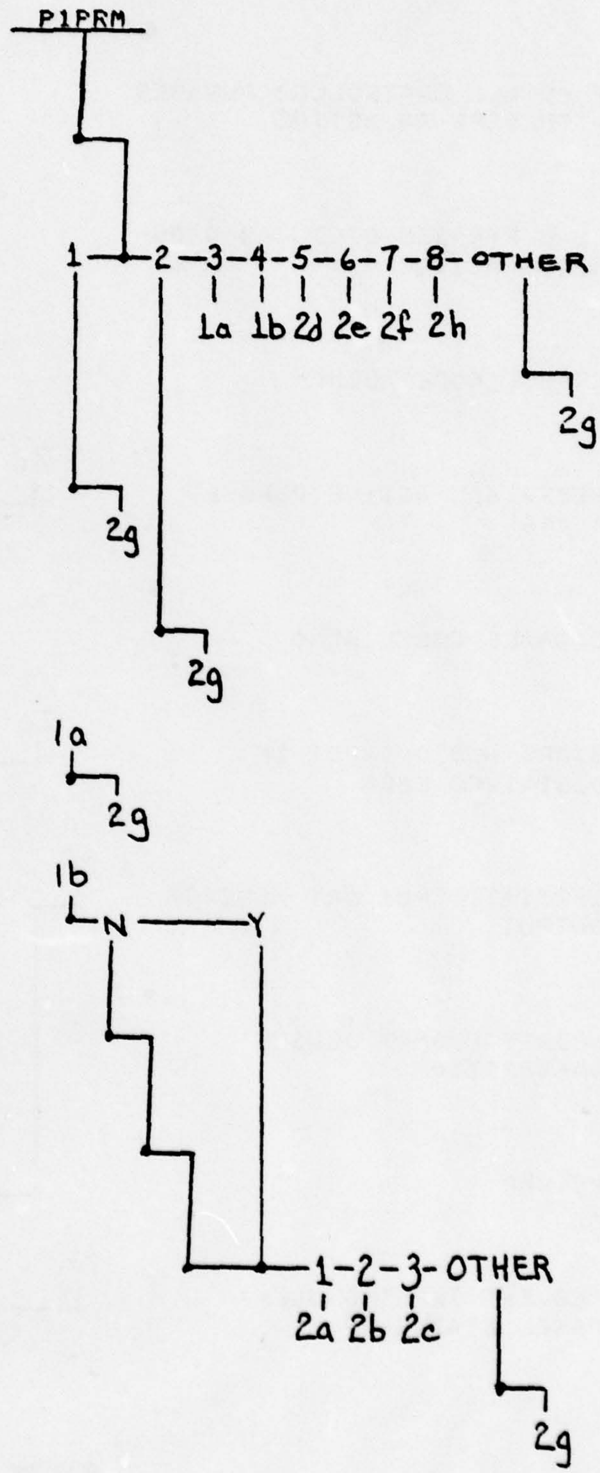
IS DEVICE NUMBER LEGAL?

TYPE "PRMT DEV ERR."

SET DEVICE TO CRT

BRANCH ON PHRASE SET

TYPE "PRMT SET ERR."



P1PRM FLOWCHART (SHEET 1 OF 2)

FLAG ALL CONTROLLER PHRASES AS ACTIVE

FLAG ALL CONTROLLER PHRASES WITH VRPS AS ACTIVE

FLAG PHRASES GIVEN AS ARGUMENTS ACTIVE

ENABLE MODEL DEMO

RESET ALL ACTIVE PHRASE FLAGS

DISABLE MODEL DEMO

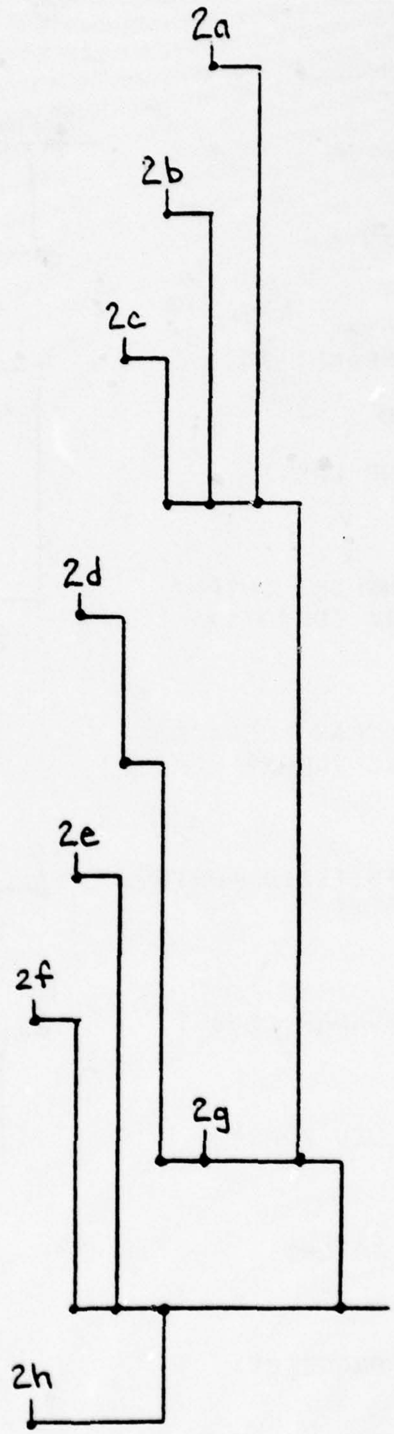
STORE AUDIO INPUT IN DIGITIZED FORM

INITIATE CPU1 CRT MESSAGE OUTPUT

UPDATE PROMPT DEVICE INFORMATION

RETURN

REQUEST INPUT STUDENT PANEL STATUS



F1PRM FLOWCHART (SHEET 2 OF 2)

P1RAD

Description: Initiates servo positioning, alignment, and activity for phase 1.

Entry point: P1RAD

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: IPBOUT1

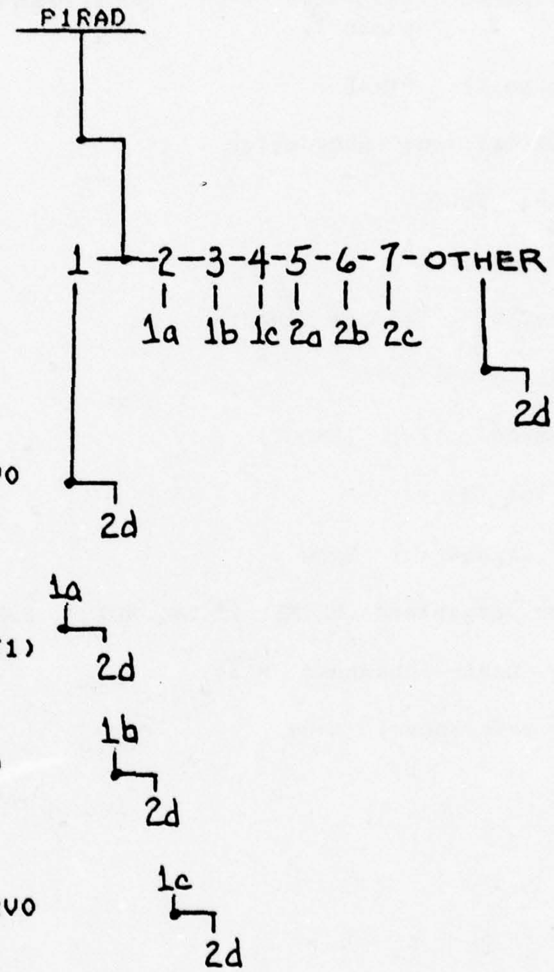
Cancels: None

Input arguments: None

Common variables: FZIFP, RDCLR, RDTDR, RDRNG, FZSRV

Files created/changed: None

Files referenced: None



READ IN RADAR ARGUMENTS

BRANCH ON IDENTIFIER

TYPE "F1 RAD ERR."

SEND ACTIVATE AZIMUTH SERVO
MESSAGE (CALL IBFOUT1)

SEND ACTIVATE ELEVATION
SERVO MESSAGE (CALL IBFOUT1)

SEND THE OFF AZIMUTH SERVO
MESSAGE

SEND THE OFF ELEVATION SERVO
MESSAGE

F1RAD FLOWCHART (SHEET 1 OF 2)

SEND THE SET AZIMUTH
ALIGNMENT MESSAGE

SET AZIMUTH VARIABLE

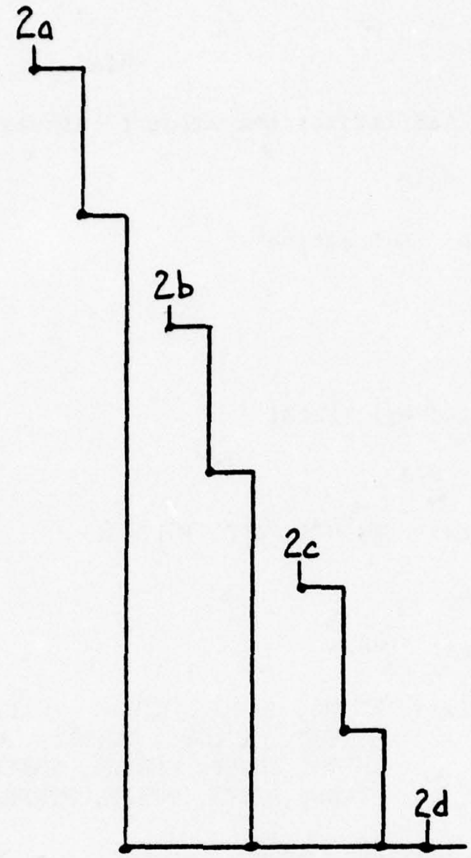
SEND THE SET ELEVATION
ALIGNMENT MESSAGE

SET ELEVATION VARIABLE

SEND THE SET RANGE
ALIGNMENT MESSAGE

SET RANGE VARIABLE

RETURN



FIRAD FLOWCHART (SHEET 2 OF 2)

P1AC

Description: Initializes the aircraft simulation variables for phase 1.

Entry point: P1AC

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: IPBOUT1, APE, P1FRZAC

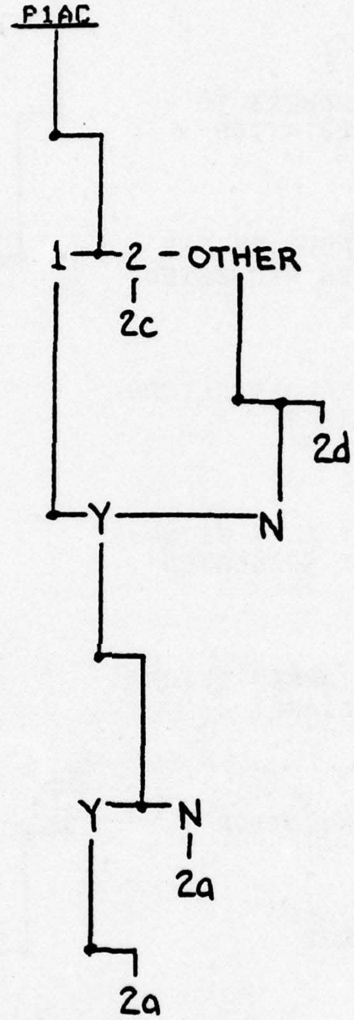
Cancels: None

Input arguments: None

Common variables: FZINP, CTCLR, CTHOF, ACALT, ACRNG, ACOFF, ACHDG, ACTYP, ACGYRO, ACICE, ACHYDR, ACENG, EMGYFL, EMGYR, EMICFL, EMICR, ENWSP, ENWHDG, ENFLUCT, PTYP, PTFLT, PTAPR, PTLOW, PTLZN, PTUZN, PTAPR, PTRNG, CTNGR, CTREL

Files created/changed: None

Files referenced: None



READ IN ARGUMENTS

BRANCH ON AIRCRAFT SIMULATION INSTRUCTION

TYPE "P1 AC ERR."

DO FIRST CARD AND SECOND CARD ID'S CHECK?

INITIALIZE FLIGHT VARIABLES AND SET INITIALIZATION FLAG

IS THE A/C POSITION SPECIFIED BY ZONE?

CONVERT ZONES TO FEET/MILES

P1AC FLOWCHART (SHEET 1 OF 2)

IS THE FLIGHT PATH ZONE
RESTRICTED?

CONVERT ZONE NUMBERS TO
INTERNAL ZONE NOTATION

HAS AZIMUTH TARGET RETURN
SUPPRESSION BEEN REQUESTED?

SEND OUT TURN TARGET RETURN
PICTURE OFF MESSAGE

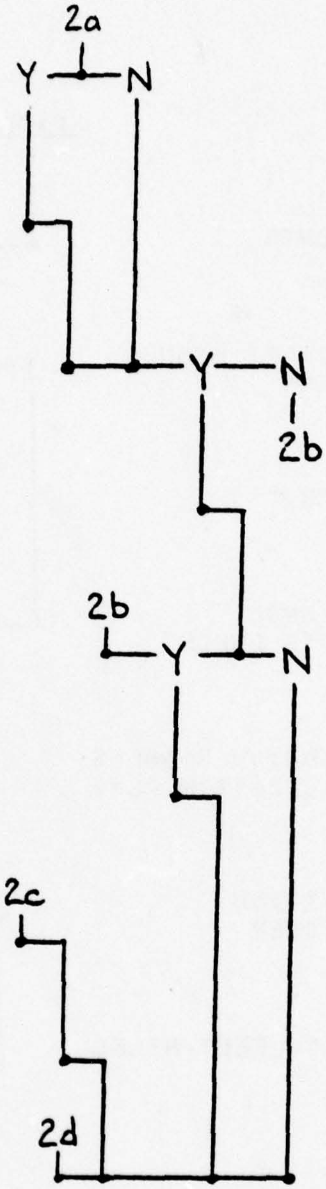
HAS ELEVATION TARGET RETURN
SUPRESSION BEEN REQUESTED?

SEND OUT TURN TARGET RETURN
PICTURE OFF MESSAGE

TURN ON A/C SIMULATION

TASK FREEZE EVENT

RETURN



P1AC FLOWCHART (SHEET 2 OF 2)

P1FRZAC

Description: Freezes simulation on specified event.

Entry point: P1FRZAC

Classification: Task

Period: None

Language: F

Activated/called by: P1AC

Cancelled by: self

Activates/calls: None

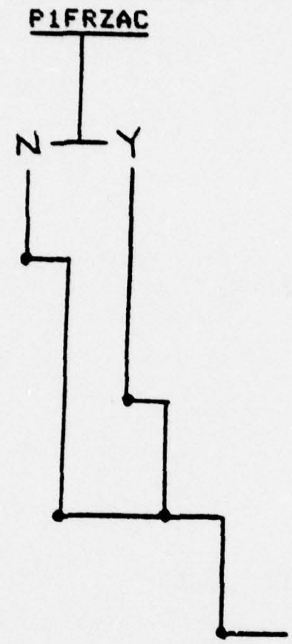
Cancels: APE

Input arguments: None

Common variables: FZFRZ, BXGLB

Files created/changed: None

Files referenced: None



IS EVENT SPECIFIED?

WAIT FOR THE FREEZE
ON ENDING RANGE

WAIT FOR EVENT SPECIFIED

KILL A/C SIMULATION

DIE

P1FRZAC FLOWCHART (SHEET 1 OF 1)

P1WAI

Description: Initiates wait conditions for phase 1.

Entry point: P1WAI

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: P1TMT, P1WAT, P1AZLR

Cancels: None

Input arguments: None

Common variables: FZINP, FZSKP

Files created/changed: None

Files referenced: None

READ IN WAIT ARGUMENTS

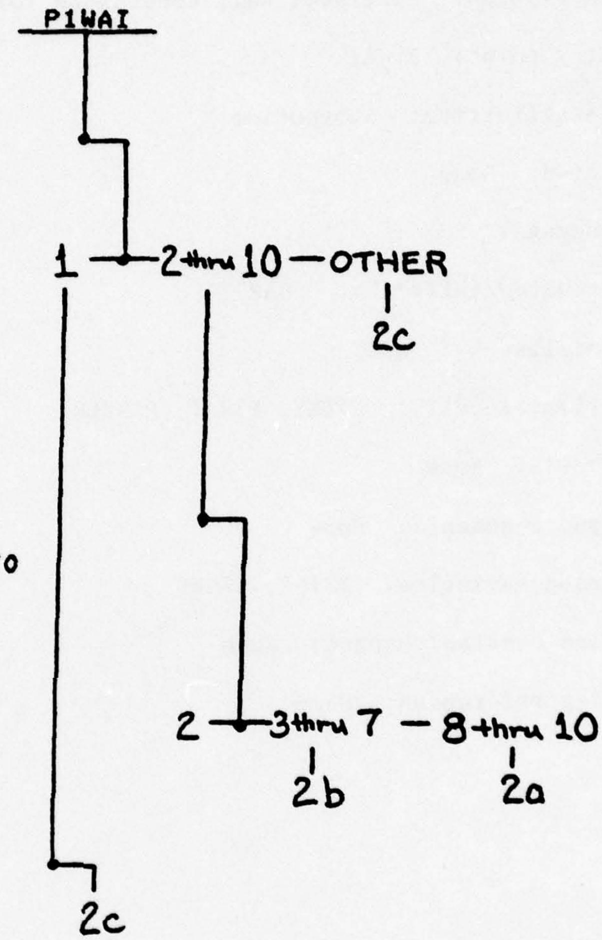
BRANCH ON WAIT TYPE

TYPE "P1 WAI ERR."

SET UP USER CLOCK WHICH TIMEOUTS ON GIVEN INPUT VALUE. TRANSFER IS MADE TO P1TMT UPON TIMEOUT.

BRANCH ON WAIT TYPE

WAIT PRESCRIBED DELAY



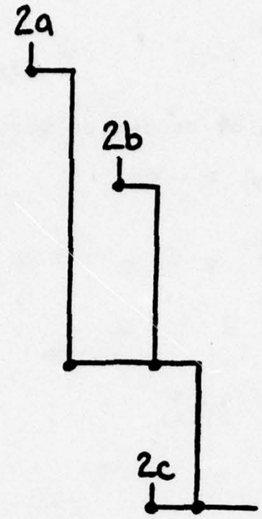
P1WAI FLOWCHART (SHEET 1 OF 2)

WAIT FOR EVENT

QUEUE A .5 SEC PERIODIC TASK
WHICH MONITORS THE REQUESTED
VARIABLE

WAIT FOR PHASE EXECUTIVE
WAKEUP

RETURN



P1WAI FLOWCHART (SHEET 2 OF 2)

P1AZLR

Description: Waits for requested servo or aircraft zone conditions.

Entry point: P1AZLR

Classification: Task

Period: 0.5 second

Language: F

Activated/called by: P1WAI

Cancelled by: P1TMT

Activates/calls: None

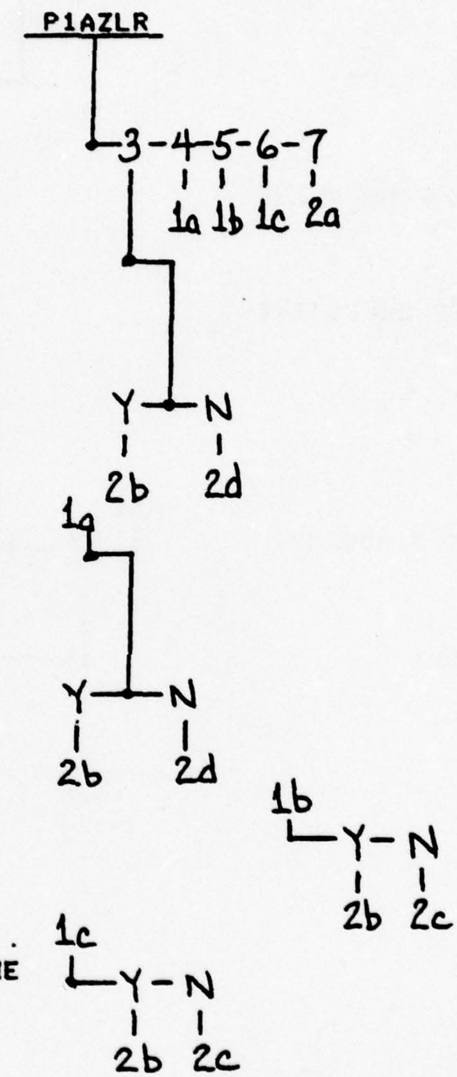
Cancels: User clock

Input arguments: None

Common variables: FZINP, RDSVAZ, RDSVEL, ACRNG, ACAZN, ACEZN

Files created/changed: None

Files referenced: None



BRANCH ON WAIT TYPE

TRANSLATE AZIMUTH SERVO ZONE TO X,Y COORDINATE

DOES THE COORDINATE MATCH THE SERVO?

TRANSLATE ELEVATION SERVO ZONE TO X,Y COORDINATE

DOES THE COORDINATE MATCH THE SERVO?

DOES THE A/C AZIMUTH ZONE MATCH?

DOES THE A/C ELEVATION ZONE MATCH?

P1AZLR FLOWCHART (SHEET 1 OF 2)

DOES THE A/C RANGE MATCH?

REMOVE USER CLOCK

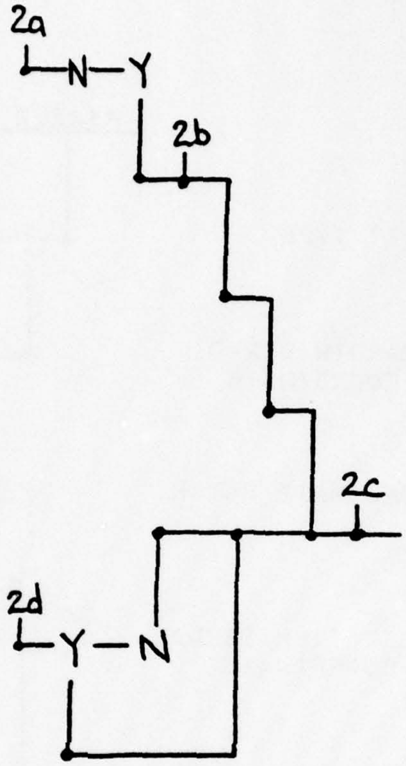
REMOVE TASK FROM QUEUE

WAKEUP PHASE EXECUTIVE

RETURN

HAS THE SERVO MOVED?

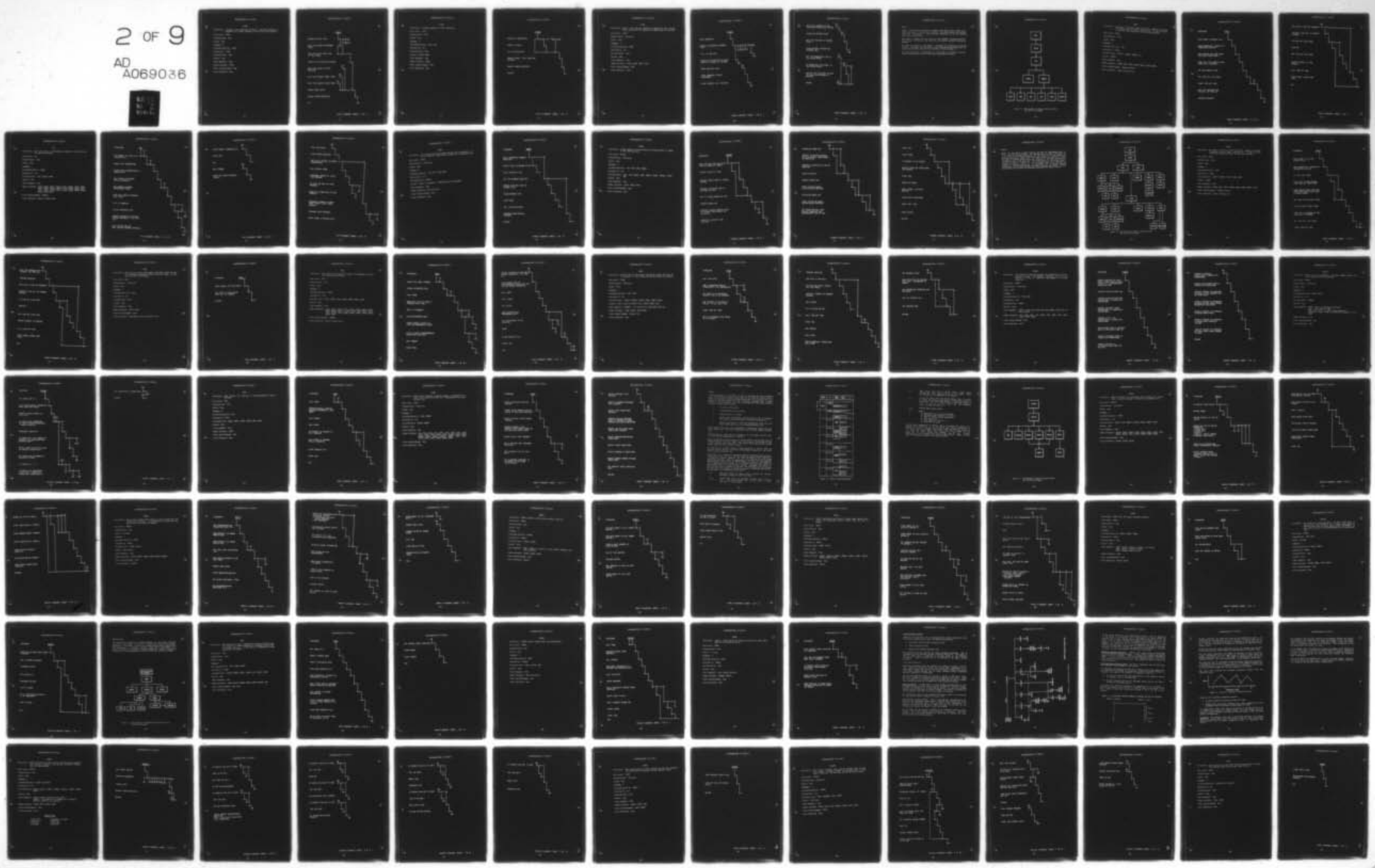
FLAG MOVEMENT



P1AZLR FLOWCHART (SHEET 2 OF 2)

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

2 OF 9
AD
A069036



P1WAT

Description: Initiates a wait condition for phase 1. The wait condition is dependent on a student response or the Votrax output complete.

Entry point: P1WAT

Classification: Task

Period: None

Language: F

Activated/called by: P1WAI

Cancelled by: P1TMT

Activates/calls: None

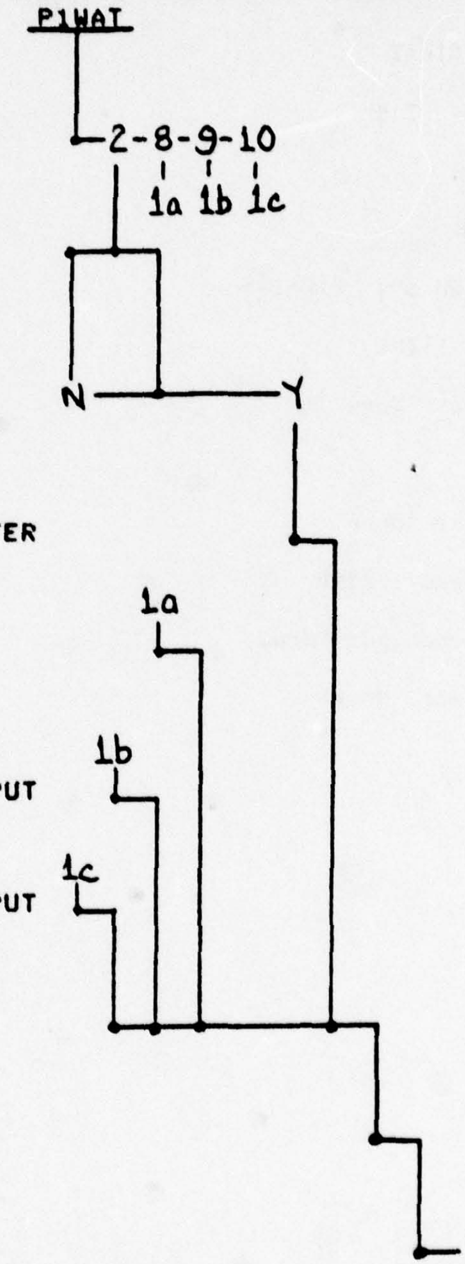
Cancels: None

Input arguments: None

Common variables: FZINP

Files created/changed: None

Files referenced: None



BRANCH ON WAIT TYPE

WAIT FOR STUDENT KEYBOARD INPUT

DOES THE ENTRY MATCH ONE IN THE TABLE?

UPDATE FILE POSITION POINTER

WAIT FOR VOTRAX TO STOP SPEAKING

WAIT FOR STUDENT PANEL INPUT

WAIT FOR STUDENT VOICE INPUT

REMOVE USER CLOCK

WAKEUP PHASE EXECUTIVE

DIE

P1WAT FLOWCHART (SHEET 1 OF 1)

P1TMT

Description: Initiates timeouts for wait conditions.

Entry point: P1TMT

Classification: Task

Period: None

Language: F

Activated/called by: User clock

Cancelled by: N/A

Activates/calls: None

Cancels: P1AZLR, P1WAT

Input arguments: None

Common variables: FZSKP

Files created/changed: None

Files referenced: None

BRANCH ON IDENTIFIER

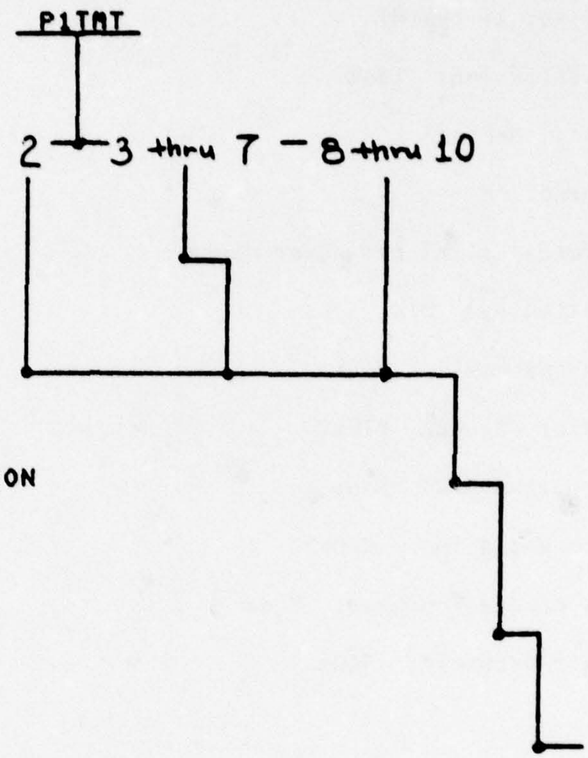
DEQUEUE P1AZLR

KILL P1AZLR/P1WAT

UPDATE PHASE 1 FILE POSITION POINTER

WAKEUP PHASE EXECUTIVE

RETURN



P1TMT FLOWCHART (SHEET 1 OF 1)

P1SEQ

Description: Phase 1 file sequence commands are handled by this routine. Skips, subroutine, and conditional command types are provided.

Entry point: P1SEQ

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ1

Cancelled by: N/A

Activates/calls: None

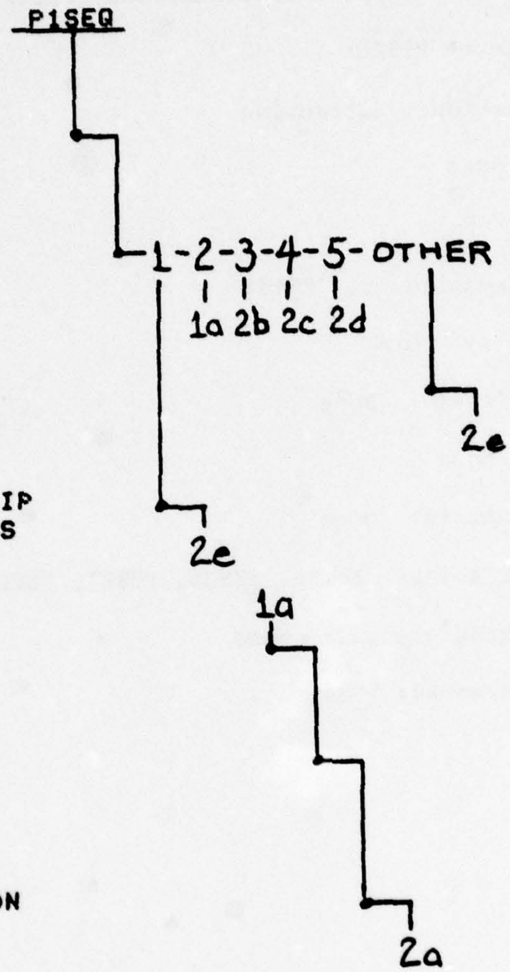
Cancels: None

Input arguments: None

Common variables: FZINP, FZSUB, FZRET, FZFLG

Files created/changed: None

Files referenced: None



READ ARGUMENTS

BRANCH ON SEQUENCE COMMAND TYPE

TYPE "P1 SEQ ERR."

UPDATE FILE POSITION TO SKIP REQUESTED NUMBER OF RECORDS

CHECK NESTING LEVEL

STORE ABNORMAL RETURN REFERENCES

STORE PRESENT FILE POSITION

P1SEQ FLOWCHART (SHEET 1 OF 2)

SET FILE POINTER TO THE
REQUESTED SUBROUTINE RECORD

CALCULATE RETURN POINT

SET FILE POINTER TO RETURN
POINT

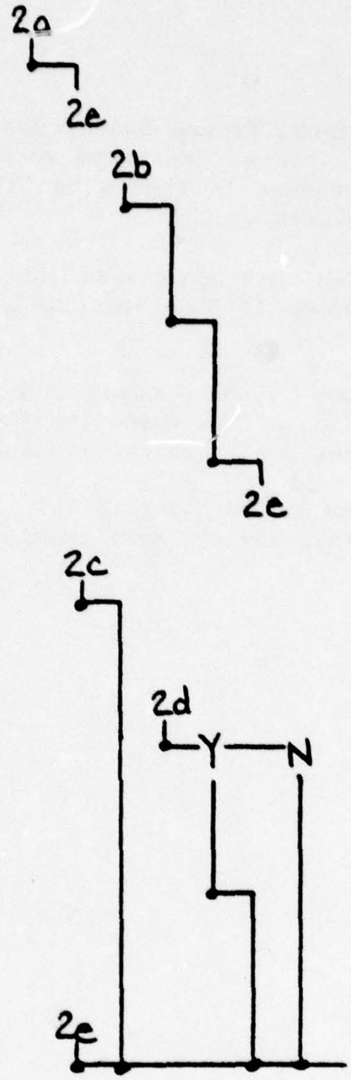
CLEAR RETURN INFORMATION
STORAGE AREA

SET THE REQUESTED FLAG TO
THE VALUE GIVEN

IS REQUESTED FLAG EQUAL TO
THE VALUE GIVEN?

UPDATE FILE POSITION TO SKIP
THE INDICATED NUMBER OF
RECORDS

RETURN



P1SEQ FLOWCHART (SHEET 2 OF 2)

Phase 2

Phase 2 is an optional freeze and feedback mode wherein the trainee practices a new skill. If any error is made, the system freezes and explains the error. The student is then given the opportunity to try the problem again from the beginning.

The phase 2 problems are also used for some remedial training exercises. When a phase 2 problem is thus specified, no optional override privilege is granted.

As shown in figure 11, the phase 2 executive is initiated by training control. Subdivision of the executive into subprograms both modularizes the processing and makes some routines available for use in phase 3.

The phase executive is responsible for initiating the simulation routines, speech understanding, and the performance measurement routines.

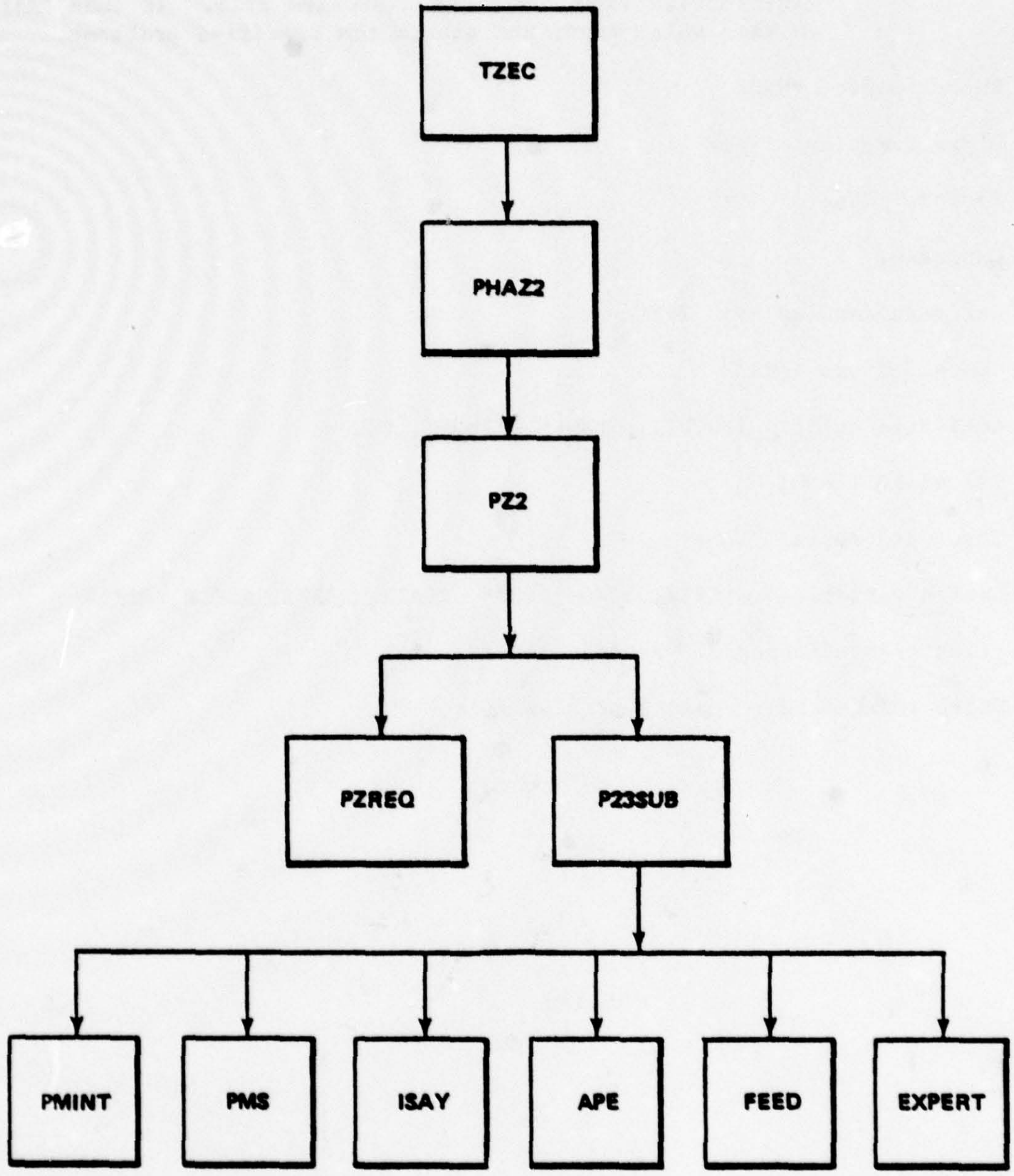


Figure 11. Relationship of Phase 2 Routines Within the GCA-CTS Environment

PHAZ2

Description: The phase 2 executive reads the initial comments and header information from the phase 2 problem file. It then initiates a task which reads and starts the specified problems.

Entry point: PHAZ2

Classification: Task

Period: None

Language: F

Activated/called by: TZEC

Cancelled by: PHAZ2

Activates/calls: IPBOUT1, SGNOFF, TIMOUT, PZ2

Cancels: TIMOUT

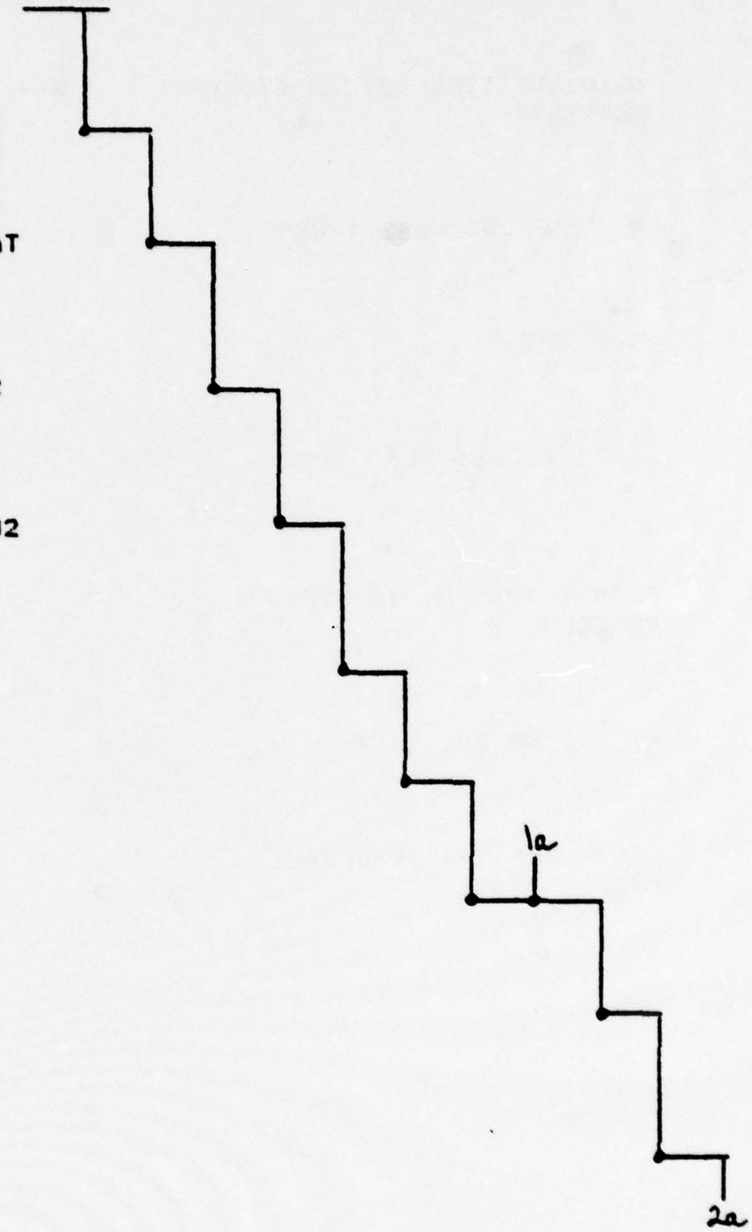
Input arguments: None

Common variables: NCPHZ, NCAO, PFS**, RDAZR, RDAZS, RDELR, RDELS

Files created/changed: Student records

Files referenced: Phase 2 problem file

PROCEDURE PHAZ2



OPEN PHASE 2 PROBLEM FILE

READ COMMENT(S), DISPLAY AT INSTRUCTOR STATION

READ HEADER (SETTING RADAR AND SERVO CONDITIONS)

SEND TEXT FILE NAME TO CPU2 (ENDS WITH 'PROCEED')

SET PMV FREEZE FLAGS

SET TIME OUT FLAG FALSE

START TIME OUT TASK

WAIT FOR MESSAGE FROM STUDENT, OR TIME OUT

MESSAGE RECEIVED

PHAZ2 FLOWCHART (SHEET 1 OF 2)

WAS THIS A TIME OUT MESSAGE?

INDICATE TIME OUT IN STUDENT RECORDS

IS TIME OUT FLAG TRUE?

SIGN OFF

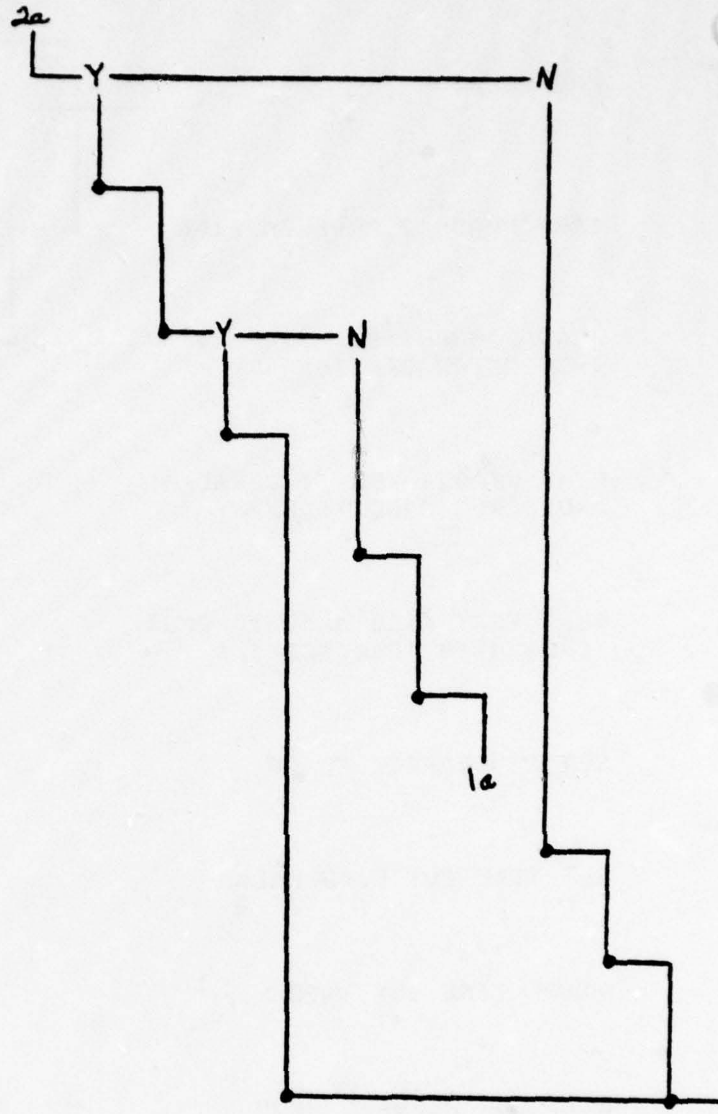
SET TIME OUT FLAG TRUE

PROMPT STUDENT TO TYPE PROCEED

KILL TIME OUT TASK

START PHASE 2 READ TASK (PZ2)

DIE



PHAZ2 FLOWCHART (SHEET 2 OF 2)

PZ2

Description: PZ2 reads phase 2 environmental simulation card sets and initiates each problem.

Entry point: PZ2

Classification: Task

Period: None

Language: F

Activated/called by: PHAZ2

Cancelled by: PZ2

Activates/calls: TZEC, P23SUB, PZREQ

Cancels: ISAY

Input arguments: None

Common variables: ACALT, ACRNG, ACOFF, ACHDG, ACTYP, ACGYRO, CTCLR, NCPHZ, EMGYR, ENWSP, ENWHDG, ENFLUCT, GZNR, GZFRZ, GZTRY, PFS**, PTYP, PTFLT, PTLZN, PTUZN, PTLOW, PTAPR, PTRNG, RDAZR, RDAZS, RDELR, RDELS

Files created/changed: SCRATCH

Files referenced: Phase 2 problem files

PROCEDURE

SET NUMBER OF TRIES THIS
PORTION TO 0

CHSAV FILE INFORMATION

STORE CHSAV INFORMATION IN
SCRATCH FILE

CALL PZREQ TO PROCESS
SPECIAL REQUESTS

SET FREEZE ON ERROR
TERMINATION FALSE

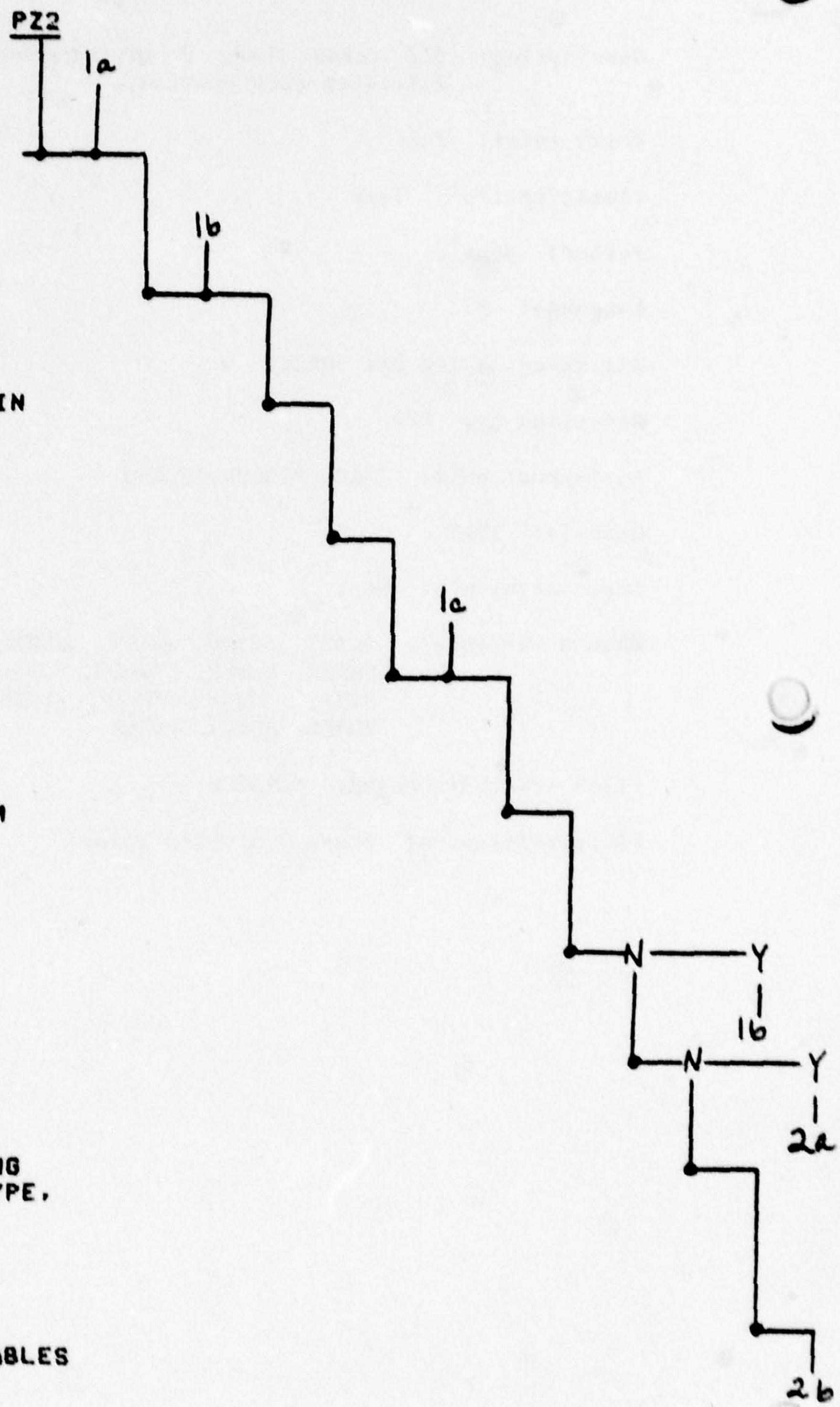
READ NEXT CARD IN PROBLEM
FILE (.RDL)

IS IT A COMMENT?

DID WE ENCOUNTER EOF?

ENCODE INFORMATION SETTING
COMMON VARIABLES: A/C TYPE,
ETC.

READ SECOND CARD OF
SET, SETTING COMMON VARIABLES



PZ2 FLOWCHART (SHEET 1 OF 3)

2a

CLOSE PHASE 2 PROBLEM FILE

START TZEC

DIE

2b

CALL P23SUB

START PMV FREEZE MODULES
(PMV##)

3a

PZ2 FLOWCHART (SHEET 2 OF 3)

WAIT FOR EVPH2

EVENT EVPHZ RECEIVED

WAS THIS A FREEZE ON ERROR
TERMINATION?

KILL ACTIVE TASKS

INCREMENT NUMBER OF TRIES
THIS PROBELM

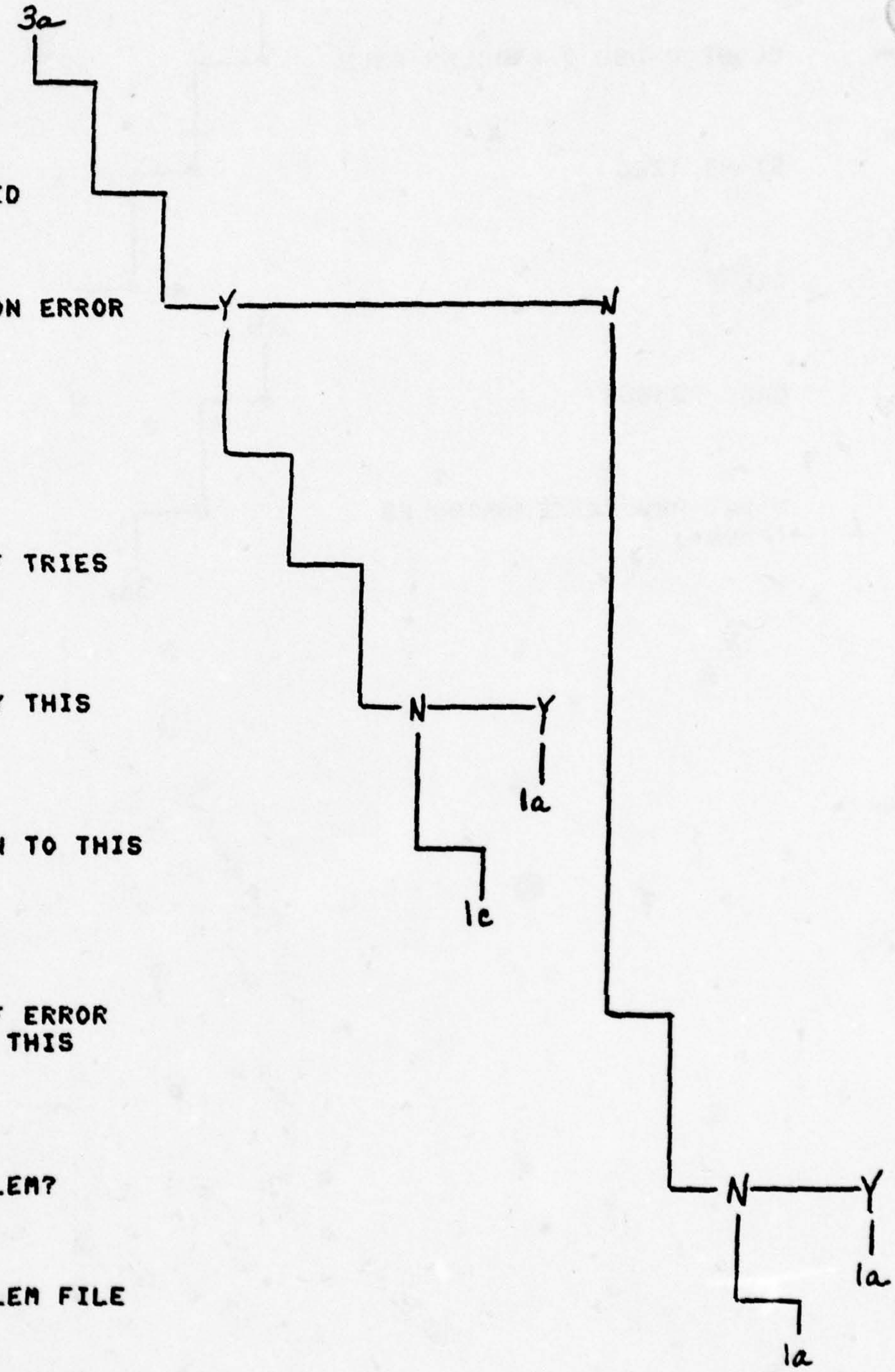
IS THIS THE 3RD TRY THIS
PROBELM?

CHRST FILE POSITION TO THIS
PROBLEM

DECREMENT NUMBER OF ERROR
FREE TRIALS TO GO, THIS
PROBELM

FINISHED THIS PROBLEM?

CHRST PHASE 2 PROBLEM FILE



PZ2 FLOWCHART (SHEET 3 OF 3)

PZREQ

Description: This routine processes special requests such as terminate. It can be called by a phase monitor routine at a convenient time.

Entry point: PZREQ

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ2, PZ3A, PZ3B, TZEC

Cancelled by: N/A

Activates/calls: IPBOUT1

Cancels: Caller if terminate or demonstration was requested

Input arguments: None

Common variables: Menu selections

Files created/changed: None

Files referenced: None

PROCEDURE **PZREQ**

WAS A TERMINATE REQUEST RECEIVED?

NOTIFY CPU2 TO RETURN TO CLI

STOP, RETURN TO CLI

DID THE STUDENT SIGN OFF?

REPORT SIGN OFF TIME IN STUDENT FILE

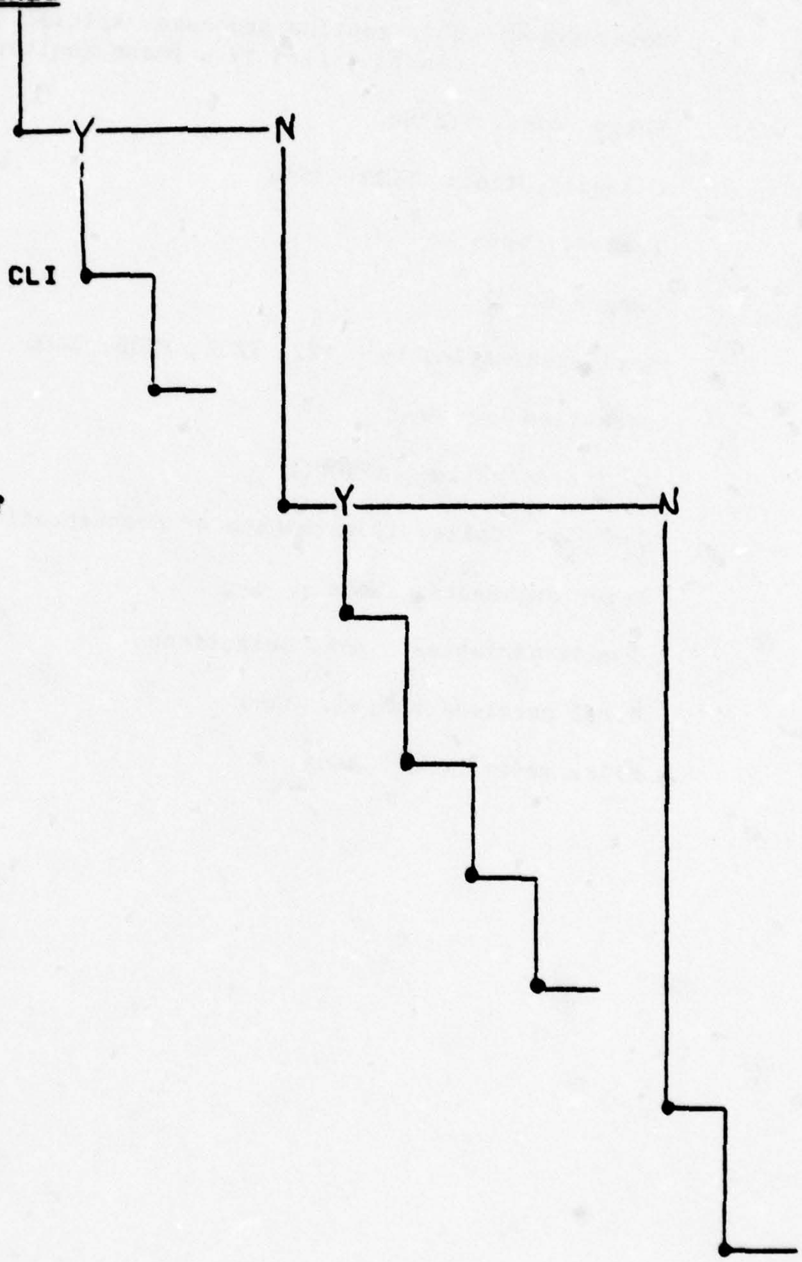
CLOSE STUDENT FILE

START DEMO

DIE, KILLING CALLER

PROCESS OTHER SPECIAL REQUESTS

RETURN



PZREQ FLOWCHART (SHEET 1 OF 1)

P23SUB

Description: P23SUB handles the initialization processing which is common to PZ2, PZ3A and PZ3B.

Entry point: P23SUB

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ2, PZ3A, PZ3B, PZDEMO

Cancelled by: N/A

Activates/calls: ISAY, APE, EXPERT, FEED, REPLAY, SPDMP, IPBIOUT, PMINT, PMS

Cancels: None

Input arguments: None

Common variables: ACOFF, ACSPD, ACALT

Files created/changed: None

Files referenced: None

PROCEDURE P23SUB

WAS STARTING POSITION SPECIFIED IN TERMS OF ZONES?

CONVERT OFFSET TO FEET

CONVERT AZMUTH ZONE TO OFFSET IN FEET

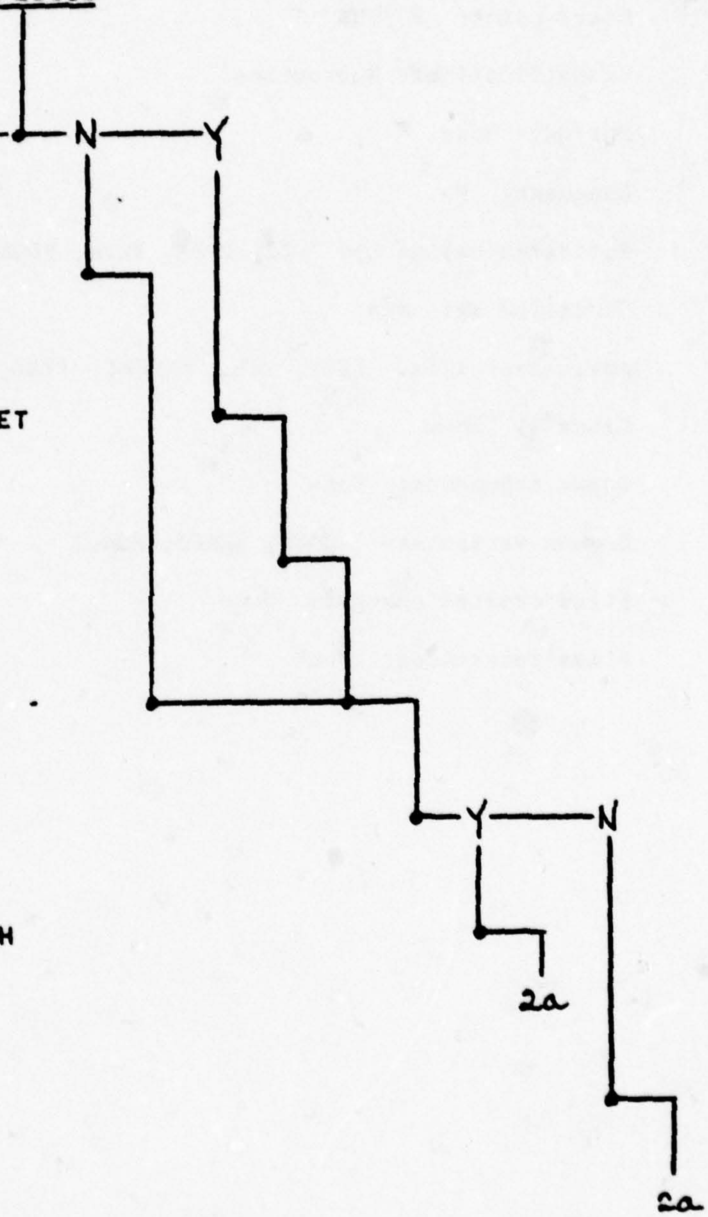
CONVERT ELEVATION ZONE TO ALTITUDE IN FEET

SET A/C SPEED BASED ON TYPE

AZIMUTH RADAR ON?

REQUEST AZIMUTH DISPLAY WITH OR WITHOUT HASH MARKS, AS SPECIFIED

RESTRICT AZIMUTH TO ON POSITION



P23SUB FLOWCHART (SHEET 1 OF 3)

ELEVATION RADAR ON?

REQUEST ELEVATION DISPLAY
WITH OR WITHOUT HASH MARKS,
AS SPECIFIED

RESTRICT ELEVATION TO THE ON
POSITION

TURN ON DISPLAY

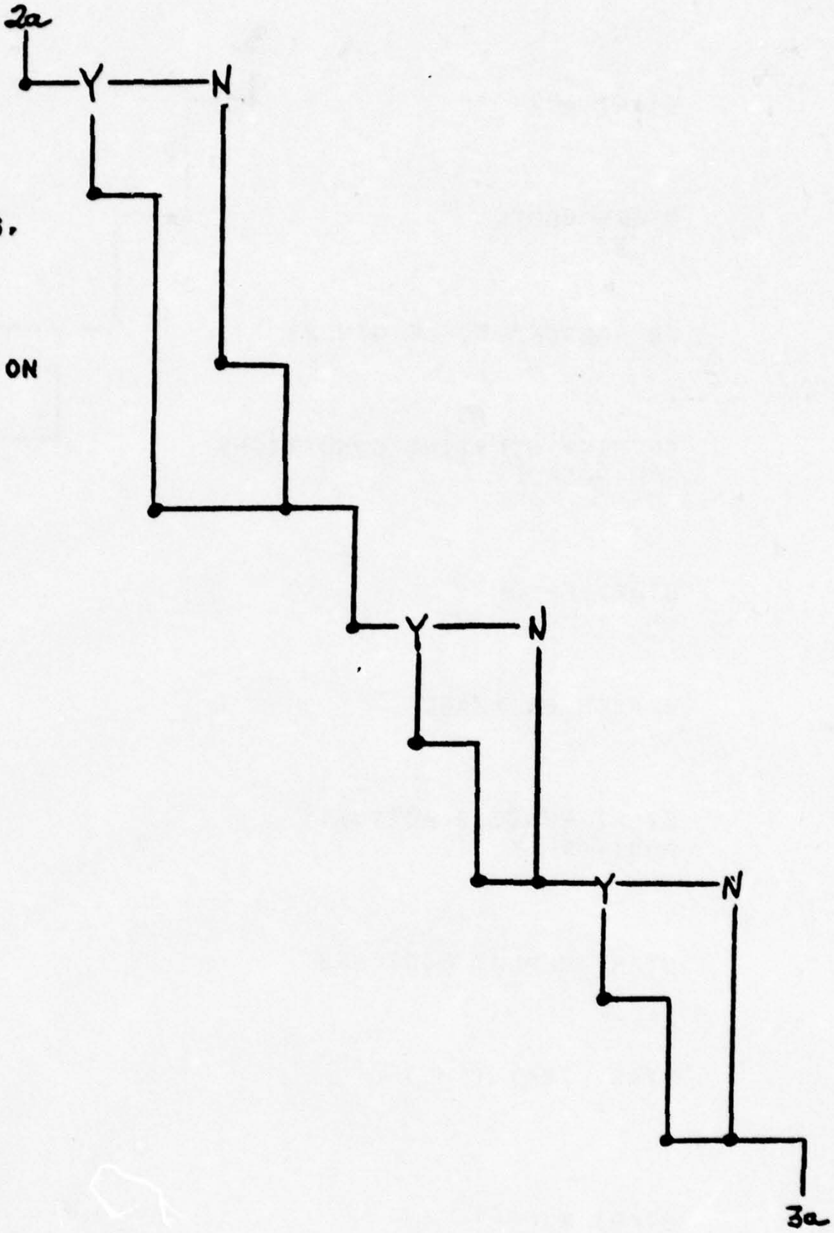
AZIMUTH SERVO ON?

START AZIMUTH SERVO,
INITIALIZING POSITION

ELEVATION SERVO ON?

START ELEVATION SERVO,
INITIALIZING POSITION

SET RANGE-RELATED TASK
STARTUP REQUESTS FOR
EMERGENCIES, END OF RUN



P23SUB FLOWCHART (SHEET 2 OF 3)

START APE

START MODEL

IS HANDOFF TO BE GIVEN?

SPECIFY STARTING CONDITIONS
FOR HANDOFF

START FEED

BRANCH ON PHASE

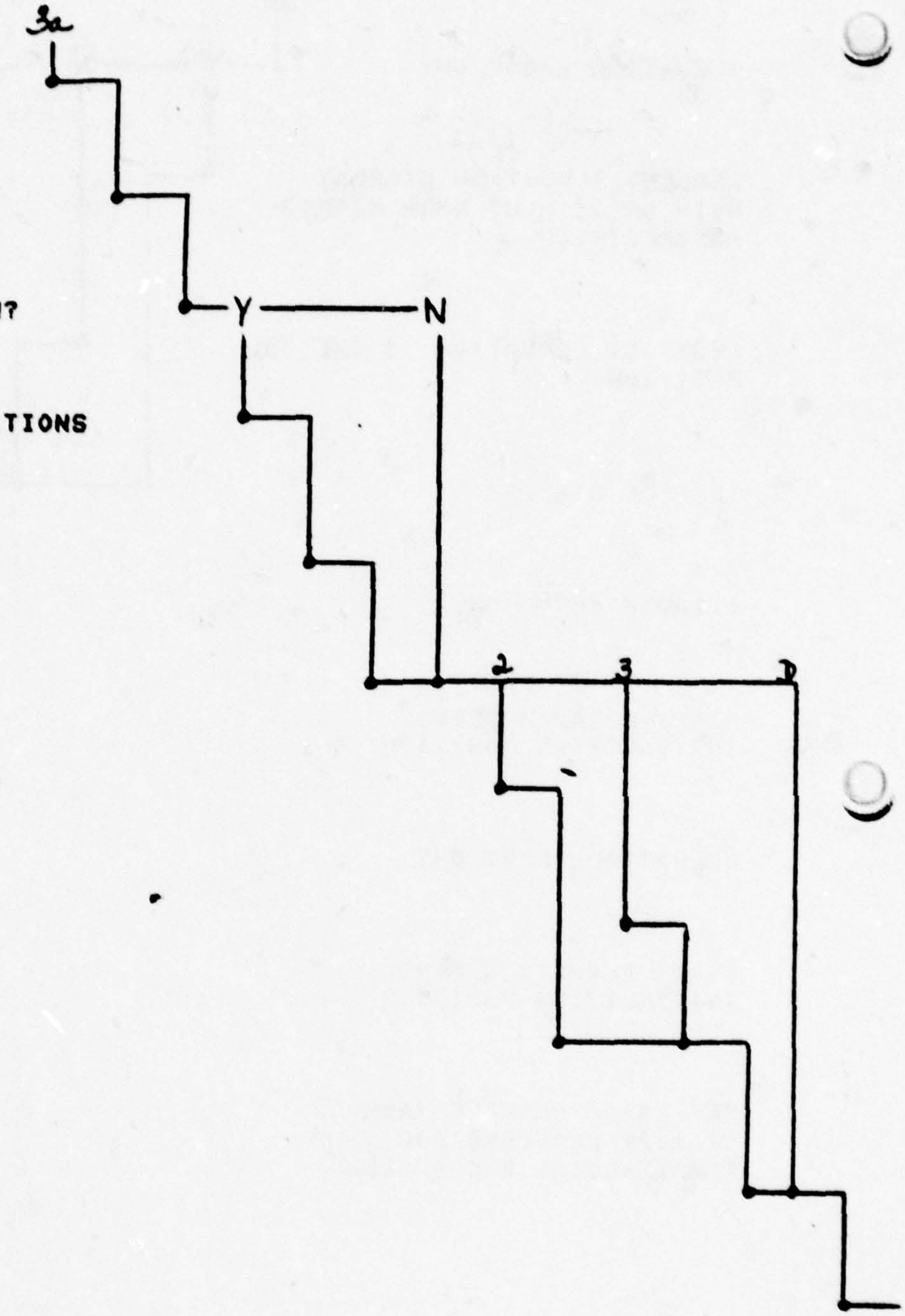
START PHASE 2 ACTIVITY
MONITOR

START REPLAY ROUTINES

START ISAY (SUS)

START EXPERT

RETURN



P23SUB FLOWCHART (SHEET 3 OF 3)

Phase 3

Phase 3 is that phase in which practice runs and the performance test, or P-run, are provided and graded. The specification of phase 3 problems can be in terms of individual problems, but the executive is also capable of choosing problems at random from a set whose limits are specified. These two modes of problem specification are reflected in the PZ3A and PZ3B controlled portions of the executive. The relationship of the various phase 3 routines is shown in figure 12. As the figure shows, this executive is invoked by training control and is responsible for initiating required simulations, speech understanding, and data collection for replay. Program descriptions for routines not previously described for phase 2 follow.

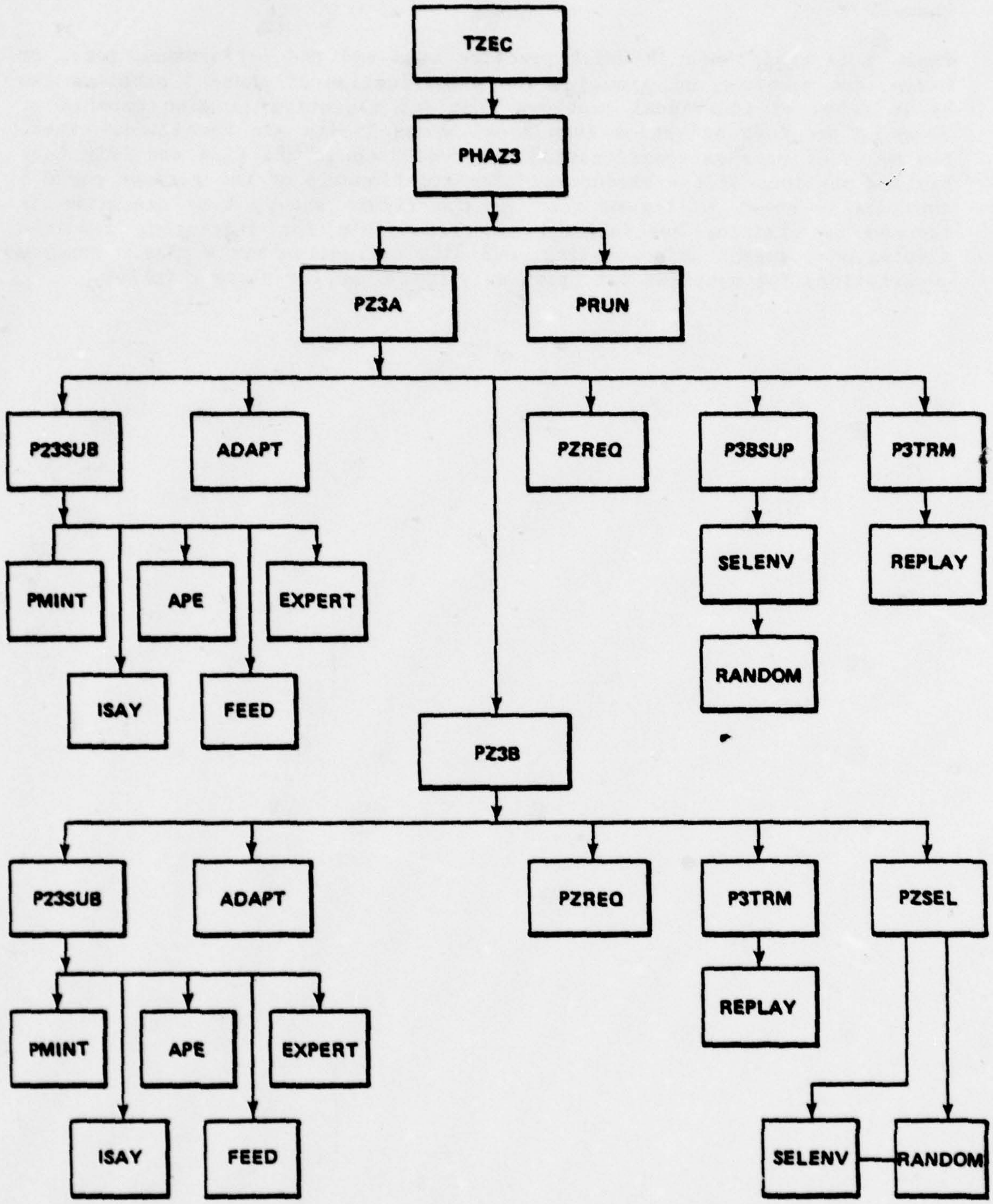


Figure 12. Relationship of Phase 3 Routines Within the GCA-CTS Environment

PHAZ3

Description: The phase 3 executive reads the initial comments and header information from the phase 3 problem file. It then initiates a task which reads and starts the specified problems.

Entry point: PHAZ3

Classification: Task

Period: None

Language: F

Activated/called by: TZEC

Cancelled by: PHAZ3

Activates/calls: IPB1OUT, SGNOFF, TIMOUT, PZ3A, PRUN

Cancels: TIMOUT

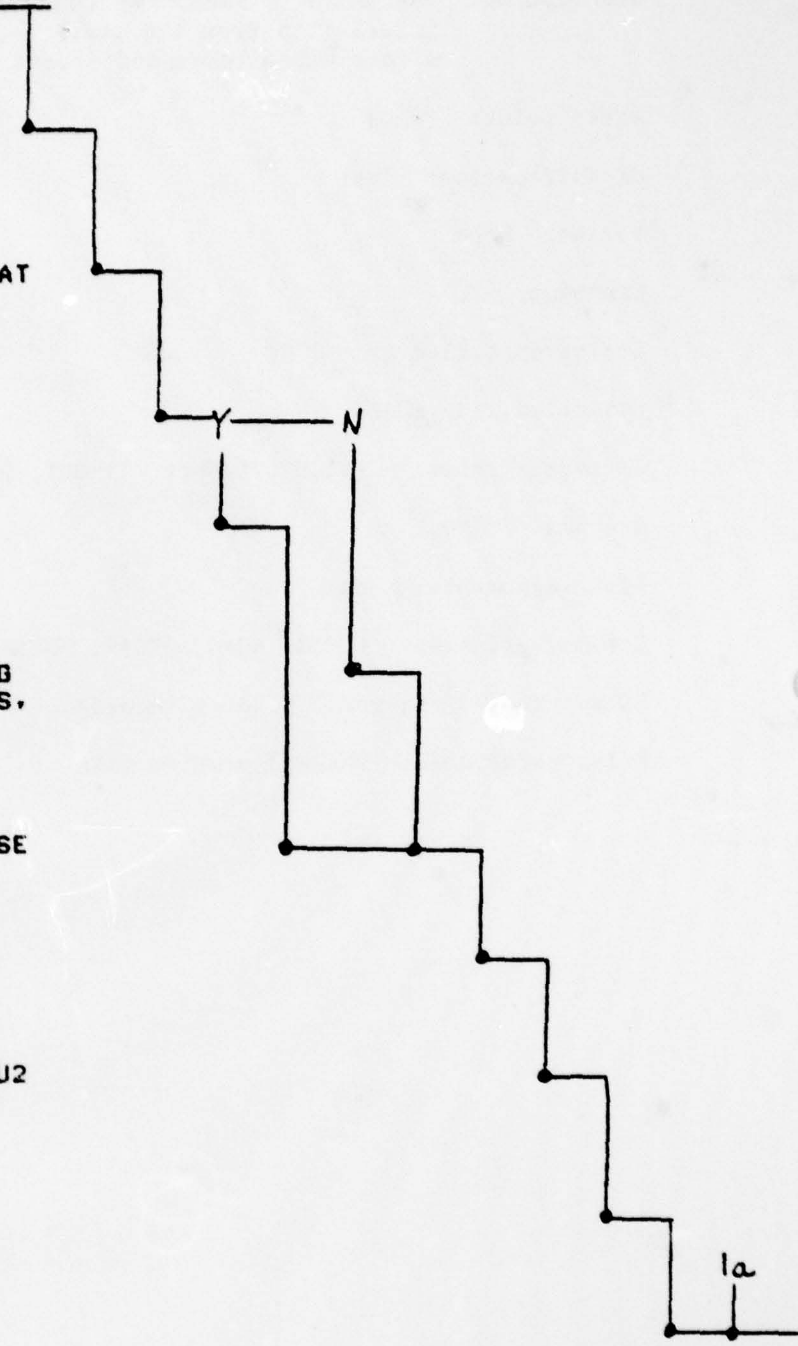
Input arguments: None

Common variables: NCPHZ, NCAO, PFS**, RDAZR, RDAZS, RDELR, RDELS, GZPAS

Files created/changed: Student records

Files referenced: Phase 3 problem file

PROCEDURE PHAZ3



OPEN PHASE 3 OR P-RUN
PROBLEM FILE

READ COMMENT(S), DISPLAY AT
INSTRUCTOR STATION

IS THIS THE P-RUN?

CALL PRUN TO READ HEADER
CARDS AND SET CONDITIONS

READ HEADER CARDS (SETTING
RADAR AND SERVO CONDITIONS,
% ERROR SCORES)

SET PASS FLAG (GZPAS) FALSE

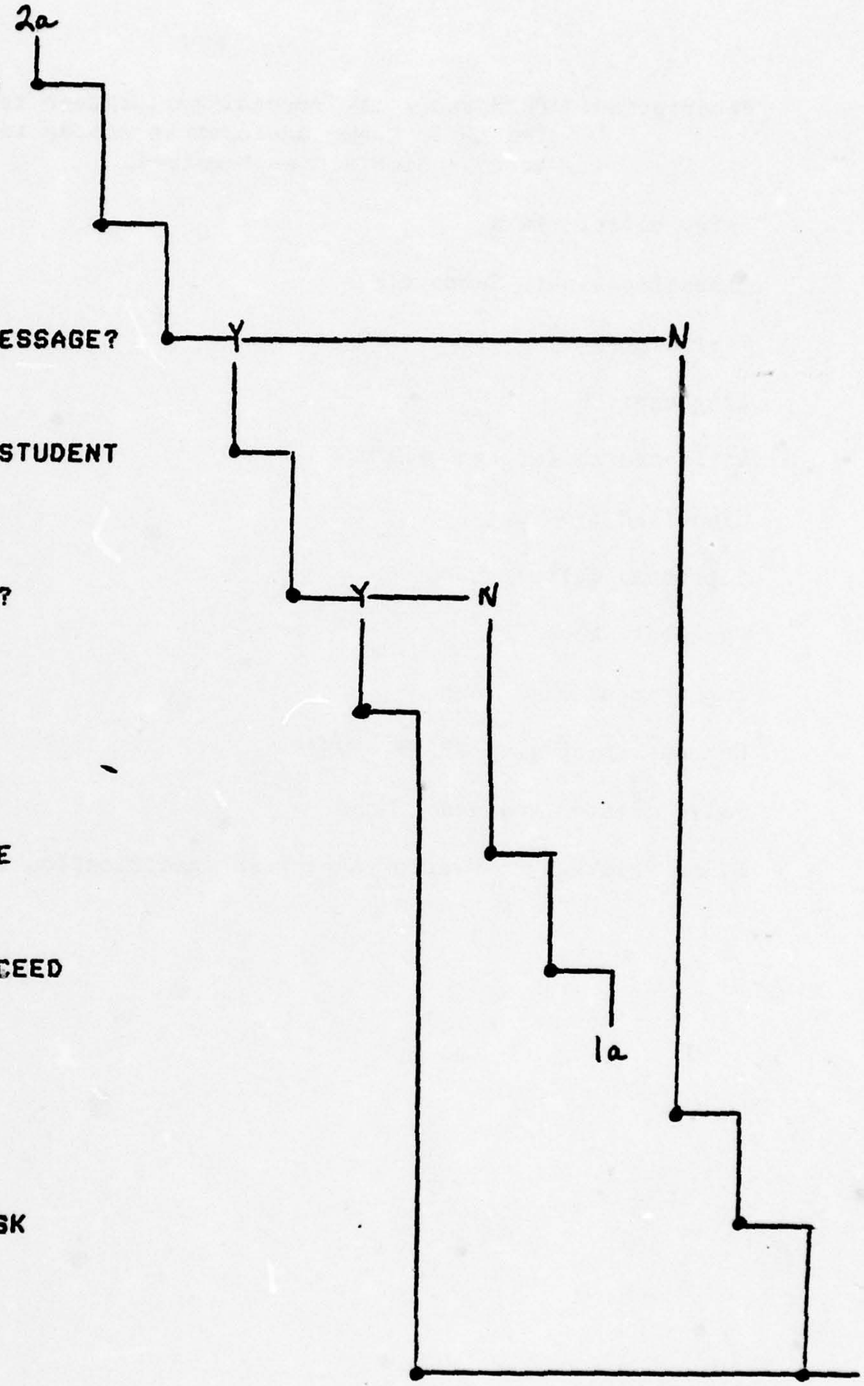
SET PMV SCORE FIRST FLAGS

SEND TEXT FILE NAME TO CPU2
(ENDS WITH 'PROCEED')

SET TIME OUT FLAG FALSE

START TIME OUT TASK

PHAZ3 FLOWCHART (SHEET 1 OF 2)



WAIT FOR MESSAGE FROM
STUDENT, OR TIME OUT

MESSAGE RECEIVED

WAS THIS A TIME OUT MESSAGE?

INDICATE TIME OUT IN STUDENT
RECORDS

IS TIME OUT FLAG TRUE?

SIGN OFF

SET TIME OUT FLAG TRUE

PROMPT STUDENT TO PROCEED

KILL TIME OUT TASK

START PHASE 3 READ TASK
(PZ3)

DIE

PHAZ3 FLOWCHART (SHEET 2 OF 2)

PRUN

Description: PRUN reads the special performance test header cards and sets the performance measurement scores required to pass. It sets other indicators as required.

Entry point: PRUN

Classification: Subroutine

Period: None

Language: F

Activated/called by: PHAZ3

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: PFS**, PVN**

Files created/changed: None

Files referenced: Performance test specification file

PROCEDURE PRUN

READ HEADER, SETTING PVN**

SET PFS** TO SHOW WHICH
PMVS WILL BE SCORED

RETURN

PRUN FLOWCHART (SHEET 1 OF 1)

PZ3A

Description: PZ3A reads single possibility phase 3 environmental card sets and initiates each problem.

Entry point: PZ3A

Classification: Task

Period: None

Language: F

Activated/called by: PHAZ3

Cancelled by: PZ3A

Activates/calls: TZEC, P23SUB, PZ3B, P3BSUP, P3TRM, PZREQ, ADAPT

Cancels: None

Input arguments: None

Common variables: ACALT, ACRNG, ACHDG, ACTYP, ACGYRO, CTCLR, NCBUG, EMGYRO, ENWSP, ENWHDG, ENFLUCT, GZPAS, PFS**, PTYP, TPFLT, PTLZN, PTUZN, PTLOW, PTAPR, PTRNG, RDAZR, RDAZS, RDELR, RDELS, NCPHZ, PVN**

Files created/changed: SCRATCH

Files referenced: Phase 3 problem files

PROCEDURE PZ3A

CHSAV FILE INFO (P3CHSV)

STORE IN SCRATCH FILE

CALL PZREQ

READ NEXT LINE OF PHASE 3
PROBLEM FILE (.RDL)

WAS IT A COMMENT?

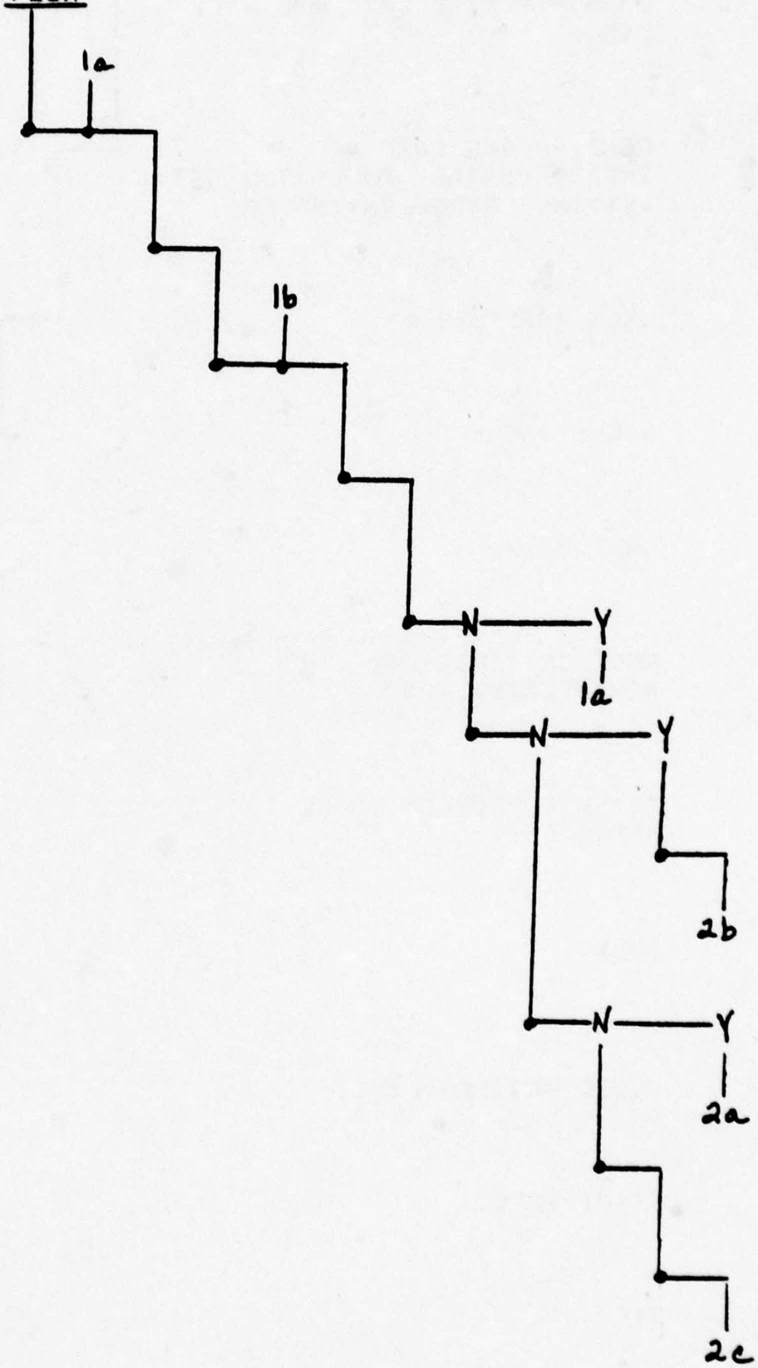
DID WE ENCOUNTER EOF?

REPORT ERROR IN BUG FILE
(TOO FEW PROBLEMS IN FILE)

IS IT A TYPE 1 ENVIRONMENTAL
SIMULATION CARD SET?

CALL P3BSUP

START PZ3B



PZ3A FLOWCHART (SHEET 1 OF 2)

ENCODE INFORMATION SETTING COMMON VARIABLES: A/C TYPE ETC

READ SECOND CARD OF ENVIRONMENTAL SIMULATION SET, SETTING COMMON VARIABLES

CALL ADAPT

CALL P23SUB

CALL P3TRM

WERE CRITERIA FOR ADVANCEMENT MET?

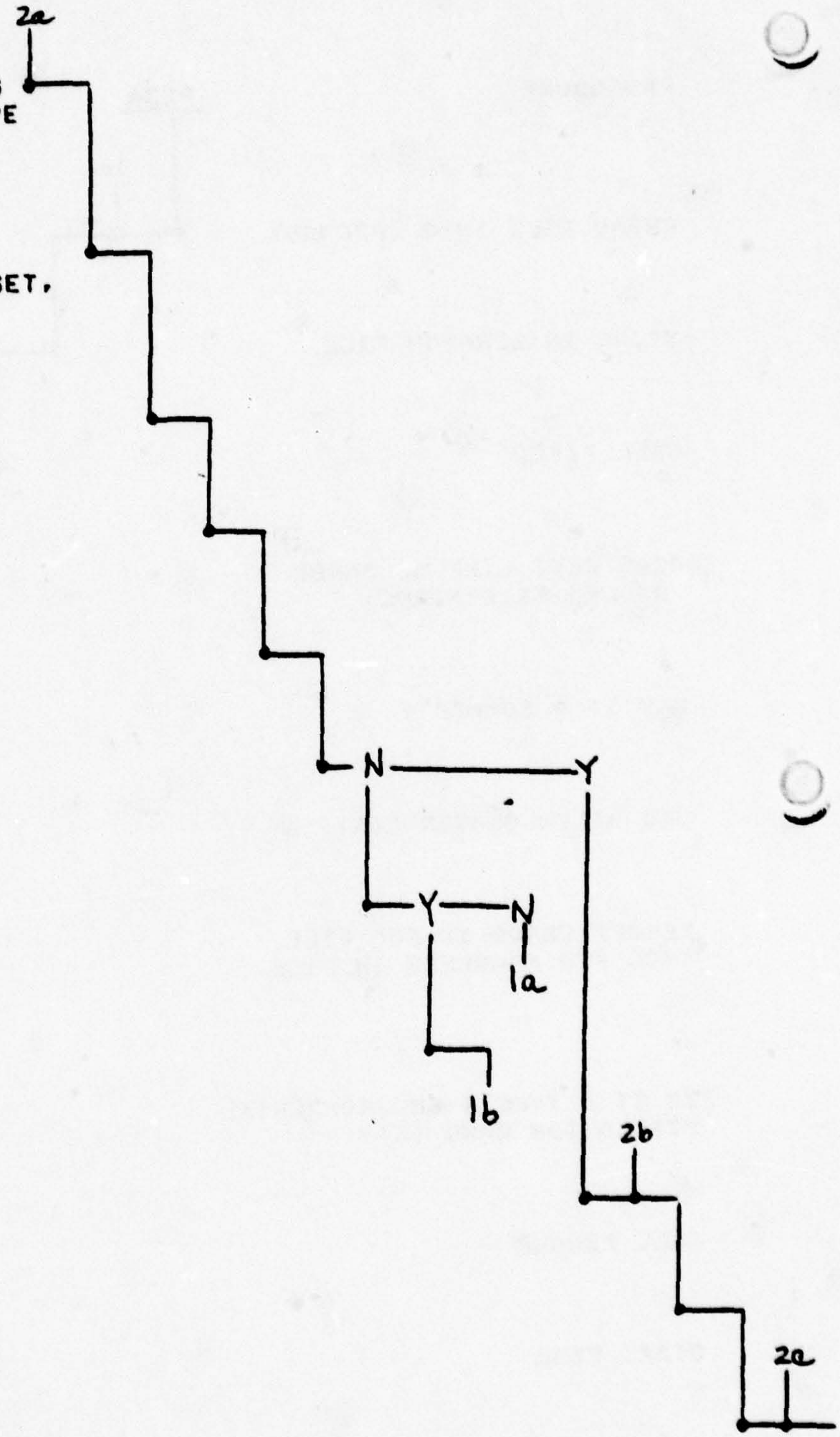
IS THIS PROBLEM TO BE REPEATED?

CHRST

CLOSE PROBLEM FILE

START TZEC

DIE



PZ3A FLOWCHART (SHEET 2 OF 2)

P3TRM

Description: P3TRM is used by both phase 3 executives to wait until the run is complete. It then invokes scoring and replay as requested.

Entry point: P3TRM

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ3A, PZ3B

Cancelled by: N/A

Activates/calls: TIMOUT, IPBOUT1, SGNOFF, SCORE, SMENU, REPLAY

Cancels: PZ3A or PZ3B on timeout only; TIMOUT, SPDMP, ISAY

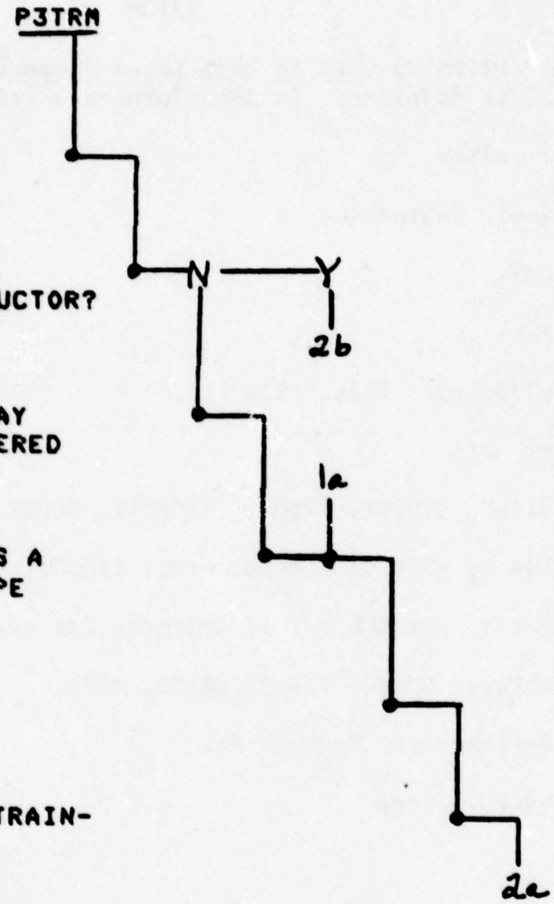
Input arguments: ADVANCE — T if criteria for advancement were met

Common variables: GZRPL, GZPAS, BXIPB, NCSR

Files created/changed: Student file

Files referenced: None

PROCEDURE



WAIT FOR EVPHZ

WAS A MANDATORY REPLAY REQUESTED BY THE INSTRUCTOR?

SET MENU TO ALLOW REPLAY REQUEST INFO TO BE ENTERED

ASK TRAINEE IF HE WANTS A REPLAY, IF SO, WHAT TYPE

START TIME OUT TASK

WAIT FOR MESSAGE FROM TRAINEE OR TIMEOUT

P3TRM FLOWCHART (SHEET 1 OF 3)

MESSAGE RECEIVED

WAS THIS A TIME OUT?

IS THIS THE FIRST TIMEOUT
FOR THIS QUERY

INDICATE TIMEOUT IN STUDENT
RECORDS

CALL SGNOFF

DIE, KILLING CALLER

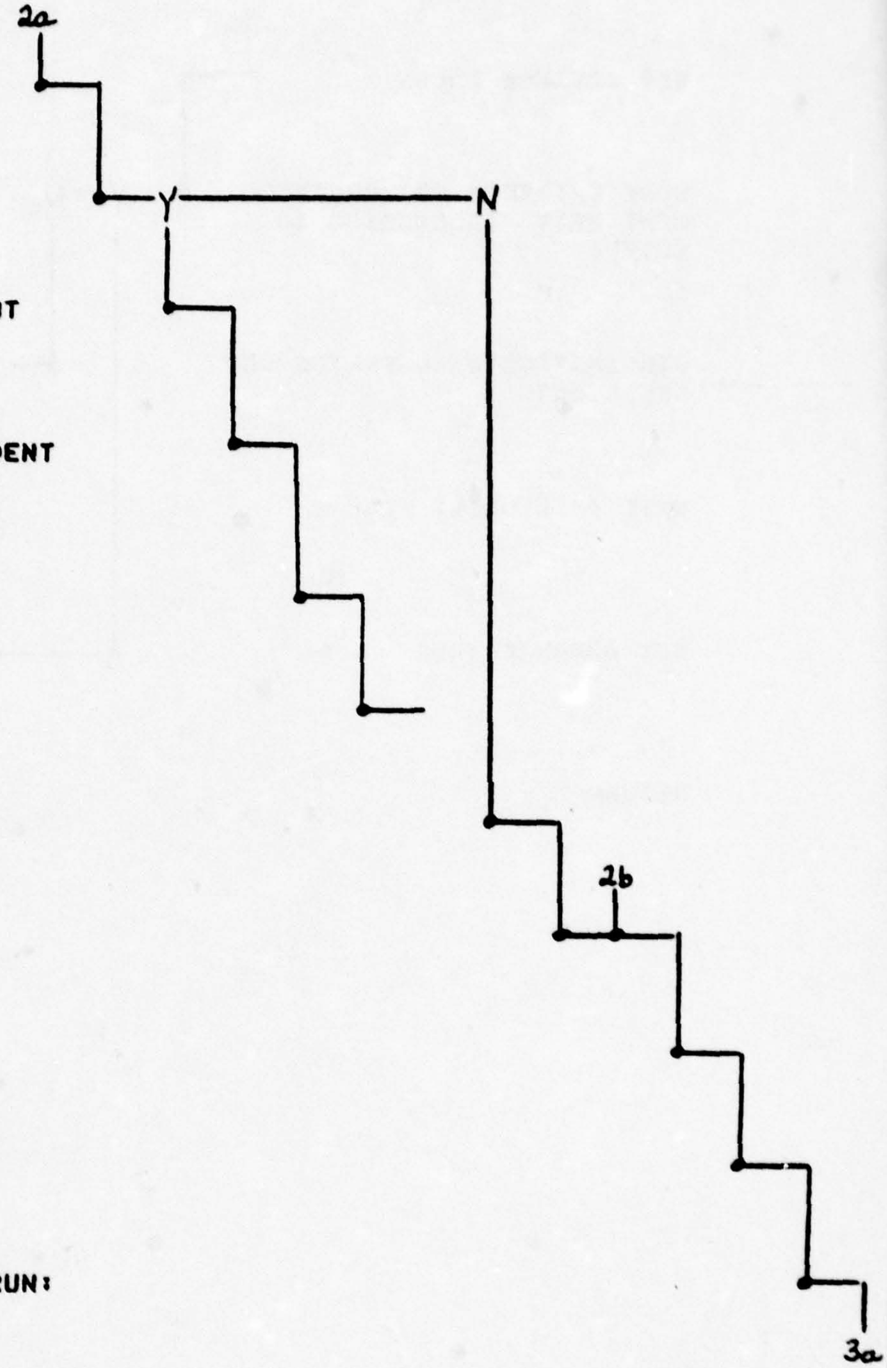
KILL TIME OUT TASK

START PMS

CALL REPLAY

WAIT EVPHZ

REPLAY COMPLETE, SCORE RUN:
CALL SCORE



P3TRM FLOWCHART (SHEET 2 OF 3)

SET ADVANCE FALSE

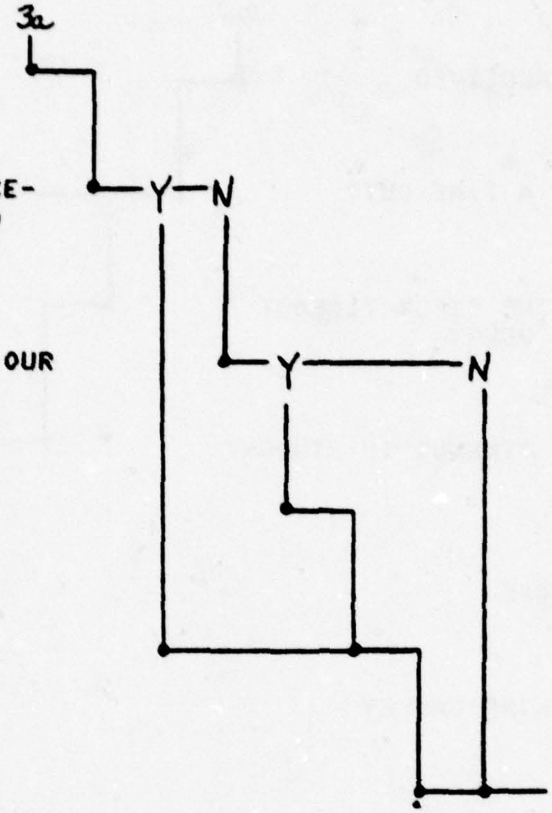
WERE CRITERIA FOR ADVANCEMENT MET? (ACCORDING TO SCORE)

DID INSTRUCTOR OVERRIDE OUR DECISION?

NOTE IN STUDENT FILE

SET ADVANCE TRUE

RETURN



P3TRM FLOWCHART (SHEET 3 OF 3)

P3BSUP

Description: This subroutine reads and decodes the information on a multi-possibility phase 3 environmental simulation card set, and converts the data to cumulative percentages for problem selection.

Entry point: P3BSUP

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ3A, DEMO

Cancelled by: N/A

Activates/calls: SELENV

Cancels: None

Input arguments: INBUF — array containing ASCII card image of the first of the two cards

Common variables: P3S, P3SV, P3SP, P3P, P3NGY, P3CL, P3WN, P3LA, P3MS, P3ICE, P3HYF, P3ENG

Files created/changed: None

Files referenced: None

PROCEDURE P3BSUP

ENCODE CONTENTS OF INPUT ARRAY (FIRST CARD OF MULTI POSSIBILITY ENVIRONMENTAL SIMULATION SET

CONVERT STARTING POSITION

CONVERT STARTING POSITION PERCENTAGES TO CUMULATIVE PERCENTAGES

CONVERT AIRCRAFT SPEED PERCENTAGES TO CUMULATIVE PERCENTAGES

CONVERT PILOT TYPE PERCENTAGES TO CUMULATIVE PERCENTAGES

READ SECOND CARD OF ENVIRON-MENTAL SIMULATION CARD SET

CONVERT APPROACH TYPES TO CUMULATIVE PERCENTAGES

COMPUTE PERCENT OF APPROACHES WHICH ARE NOT NO-GYRO

2a

P3BSUP FLOWCHART (SHEET 1 OF 2)

2a

CONVERT CLEARANCE CONDITIONS TO CUMULATIVE PERCENTAGES

CONVERT WIND CONDITIONS TO CUMULATIVE PERCENTAGES

COMPUTE PERCENT OF APPROACHES WHICH DO NOT REQUIRE LOW ALTITUDE ALERT

COMPUTE PERCENT OF APPROACHES WHICH DO NOT HAVE MINIMUM SEPARATION VIOLATIONS

COMPUTE PERCENT OF APPROACHES WHICH HAVE ICING

COMPUTE PERCENT OF APPROACHES WHICH HAVE HYDRAULIC FAILURE

COMPUTE PERCENT OF APPROACHES WHICH HAVE SINGLE ENGINE FAILURE

RETURN

P3BSUP FLOWCHART (SHEET 2 OF 2)

SELENV

Description: SELENV selects environmental conditions randomly within the limits specified for this exercise.

Entry point: SELENV

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZSEL, P3BSUP

Cancelled by: N/A

Activates/calls: RANDOM

Cancels: None

Input arguments: CPCNT - array of cumulative percentages
NELM - upper bound of CPCNT
SEED - seed value for random number generation
CONDITION - output index of condition selected out of CPCNT

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE SELENV

SET CONDITION TO 1

CALL RANDOM NUMBER GENERATOR, SUPPLYING SEED VALUE

CONVERT RANDOM NUMBER TO 0<=R<=100

IS CUMULATIVE PERCENTAGE OF ARRAY ELEMENT(CONDITION) < RANDOM VALUE R?

INCREMENT CONDITION

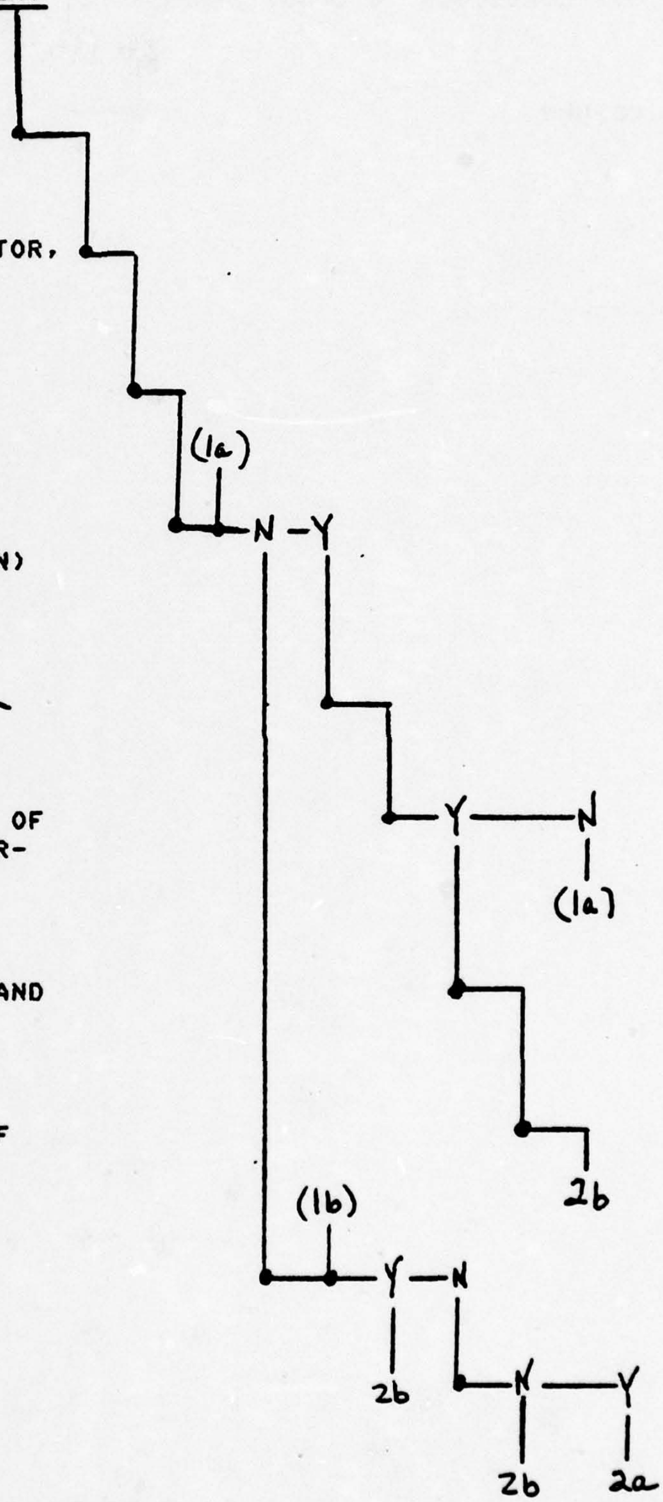
IS CONDITION > THE NUMBER OF ELEMENTS IN CUMULATIVE PERCENTAGE ARRAY?

REPORT ERROR IN BUG FILE AND ON CONSOLE (DEBUG ONLY)

SET CONDITION TO NUMBER OF ELEMENTS IN ARRAY

IS CONDITION-1 < 1?

IS CUMULATIVE PERCENTAGE (CONDITION)= CUMULATIVE PERCENTAGE (CONDITION-1)?



SELENV FLOWCHART (SHEET 1 OF 2)

2a

SET CONDITION TO CONDITION-1

(1b)

2b

RETURN

SELENV FLOWCHART (SHEET 2 OF 2)

PZ3B

Description: PZ3B controls the selection of multi-possibility phase 3 problems.

Entry point: PZ3B

Classification: Task

Period: None

Language: F

Activated/called by: PZ3A

Cancelled by: PZ3B

Activates/calls: PZSEL, ADAPT, P23SUB, P3TRM, TZEC, PZREQ

Cancels: None

Input arguments: None

Common variables: NCPHZ

Files created/changed: None

Files referenced: None

PROCEDURE

CALL PZSEL

PROCESS SPECIAL PROBLEM SELECTION REQUESTS: CALL ADAPT

CALL P23SUB

CALL P3TRM

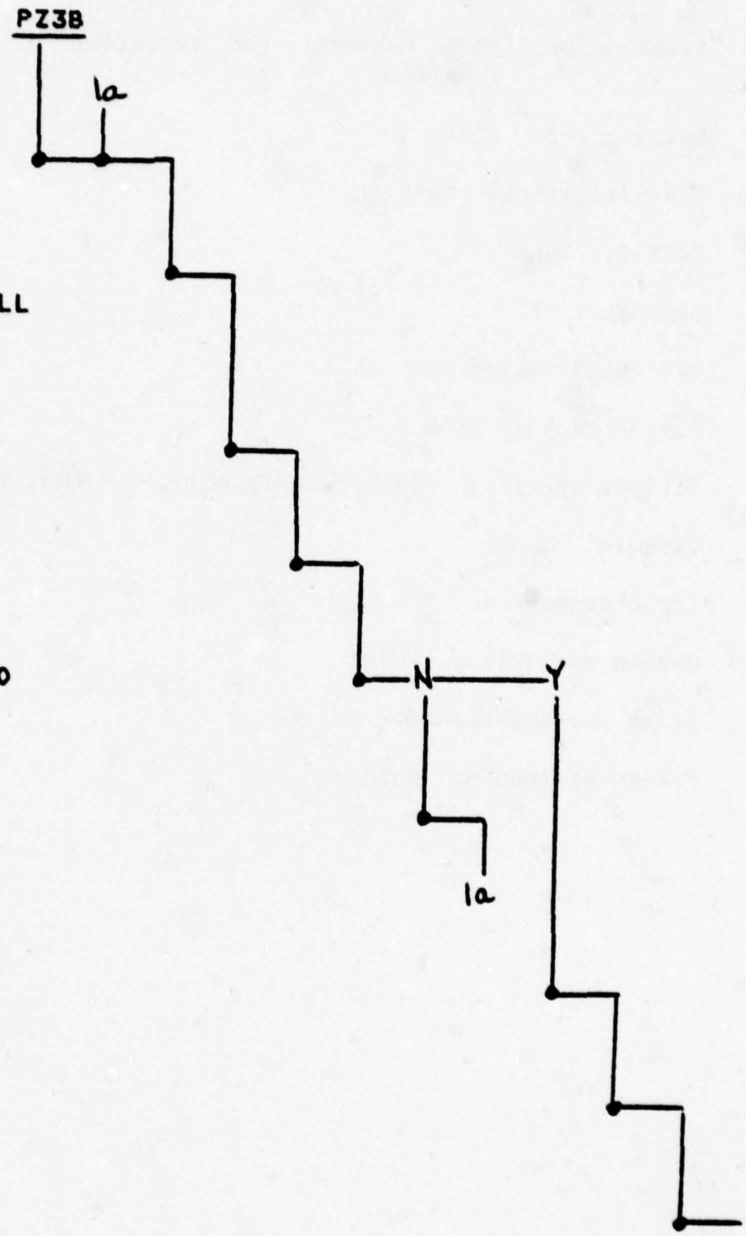
IS STUDENT TO ADVANCE TO NEXT PROBLEM?

CALL PZREQ TO PROCESS SPECIAL REQUESTS

CLOSE PROBLEM FILE

START TZEC

DIE



PZ3B FLOWCHART (SHEET 1 OF 1)

PZSEL

Description: PZSEL selects approach conditions based on information in a phase 3 multi-possibility card set or demonstration run specification file.

Entry point: PZSEL

Classification: Subroutine

Period: None

Language: F

Activated/called by: PZ3B, PZDEMO

Cancelled by: N/A

Activates/calls: SELENV, RANDOM

Cancels: None

Input arguments: None

Common variables: P3S, P3SV, P3SP, P3A, P3NGY, P3CL, P3WN, P3LA, P3MS, ACALT, ACRNG, ACOFF, ACHDG, ACTYP, ACGYRO, ACSPD, CTRLR, EMGYRO, ENWSP, ENWHDG, ENFLUCT, GZSEED, PTYP, PTFLT, PTLZN, PTUZN, PTLOW, PTAPR, PTRNG, CTHOF

Files created/changed: None

Files referenced: None

PROCEDURE

SELECT STARTING POSITION (SELENV)

CHOOSE TRACK HEADING WITHIN LIMITS OF SUPPLIED VARIATION

SELECT AIRCRAFT SPEED RANGE (SELENV)

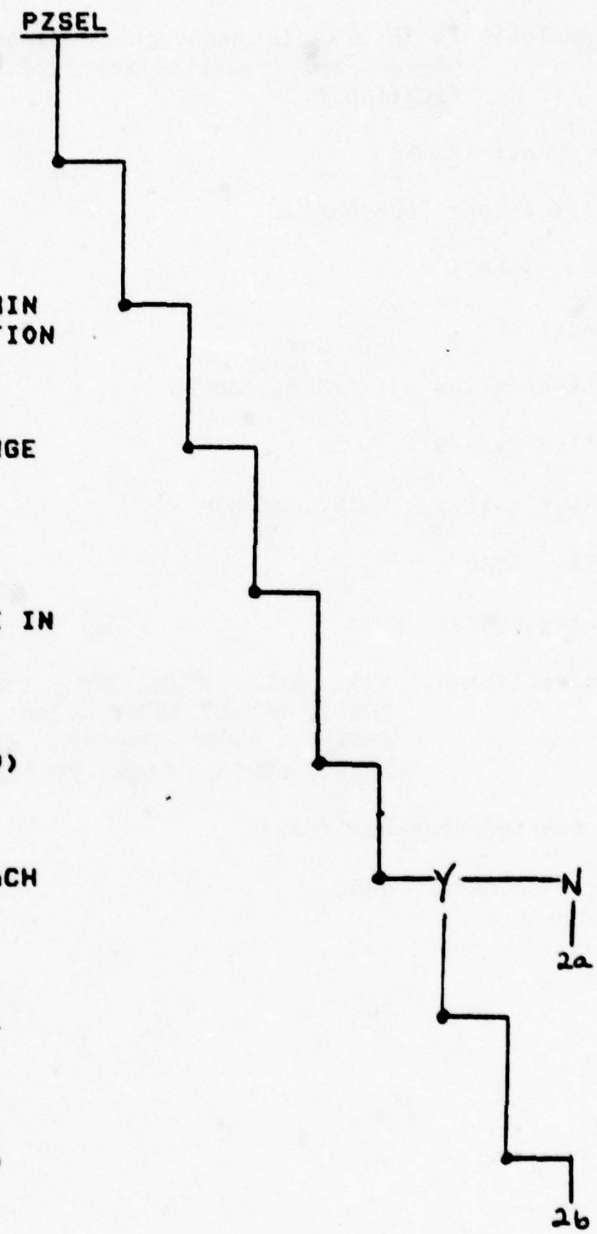
CHOOSE AIRCRAFT TYPE RANDOMLY FROM AMONG THOSE IN THE SELECTED SPEED RANGE

SELECT PILOT TYPE (SELENV)

WAS A MINIMUM FUEL APPROACH SELECTED?

SET APPROACH TYPE TO FULL STOP

SET CLEARANCE RESPONSE TO CLEARANCE TO BE GIVEN (NO W/O HERE)



PZSEL FLOWCHART (SHEET 1 OF 2)

2a

SELECT APPROACH TYPE (SELENV)

SELECT CLEARANCE RESPONSE TYPE (SELENV)

2b

SELECT WIND CONDITIONS (SELENV)

COMPUTE INITIAL HEADING BASED ON REQUESTED TRACK HEADING AND WIND CONDITIONS

SELECT LOW ALTITUDE ALERT CONDITIONS (SELENV)

SELECT MINIMUM SEPARATION CONDITIONS

SELECT ICING CONDITIONS

SELECT HYDRAULIC CONDITIONS

SELECT SINGLE ENGINE FAILURE CONDITIONS

SET HANDOFF GIVEN INDICATOR TRUE

RETURN

PZSEL FLOWCHART (SHEET 2 OF 2)

Replay

A replay capability is provided for phase 3 problems and the performance test. In all cases, scoring is done after the run using the replay files to reduce the real-time processing burden during the run, to simplify the processing of mistakes erased with the "correction" message, and to enable scoring of the performance test after misrecognitions have been resolved. There are several replay options:

1 No replay (score only)

2 Track history of the run only

3 Replay run as it occurred

4 Replay, with synthesizer repeating errors ("not on glidepath but slightly above glidepath"), and inserting omissions

5 Replay, like option 4, with rule explanations given the first time a particular error is encountered during a run.

Replay depends upon three files described in appendixes, namely the digitized speech file, the radar information file, and the student activity file.

Scoring (option 1) uses only the information in the student activity file, so it can proceed independently of replay.

Replay of the track history (option 2) causes points to be selected from the radar information and an aircraft track to be drawn on the display as fast as data transfer rates will allow. The entire track history then remains visible until the next problem is begun.

The other types of replay require double buffering of speech, radar and activity data to ensure a real-time, synchronized recreation of the events which occurred during the run.

During type 4 or 5 replay, the annotation must be coordinated with the digitized speech output in such a way that the environment freezes immediately following the incorrect behavior, the error is reported, and then normal processing continues. This is complicated by the fact that the digitized speech output is in units of one entire buffer (.5 second in this example). For the example shown in figure 13, the incorrect advisory ends within the same buffer that a correct advisory starts. Obviously, the entire t_7 speech buffer cannot be output at one time. The following outline indicates how this particular example is handled and serves to illustrate the processing described in the following string charts:

t_0: Digitized speech and radar replay routines are started. PMS begins looking for the first error.

t_{1+a}: Suppose PMS finds the incorrect advisory which ends at t_{7+c}. PMS informs SPOUT that it will have to output only part of the t_7 buffer.

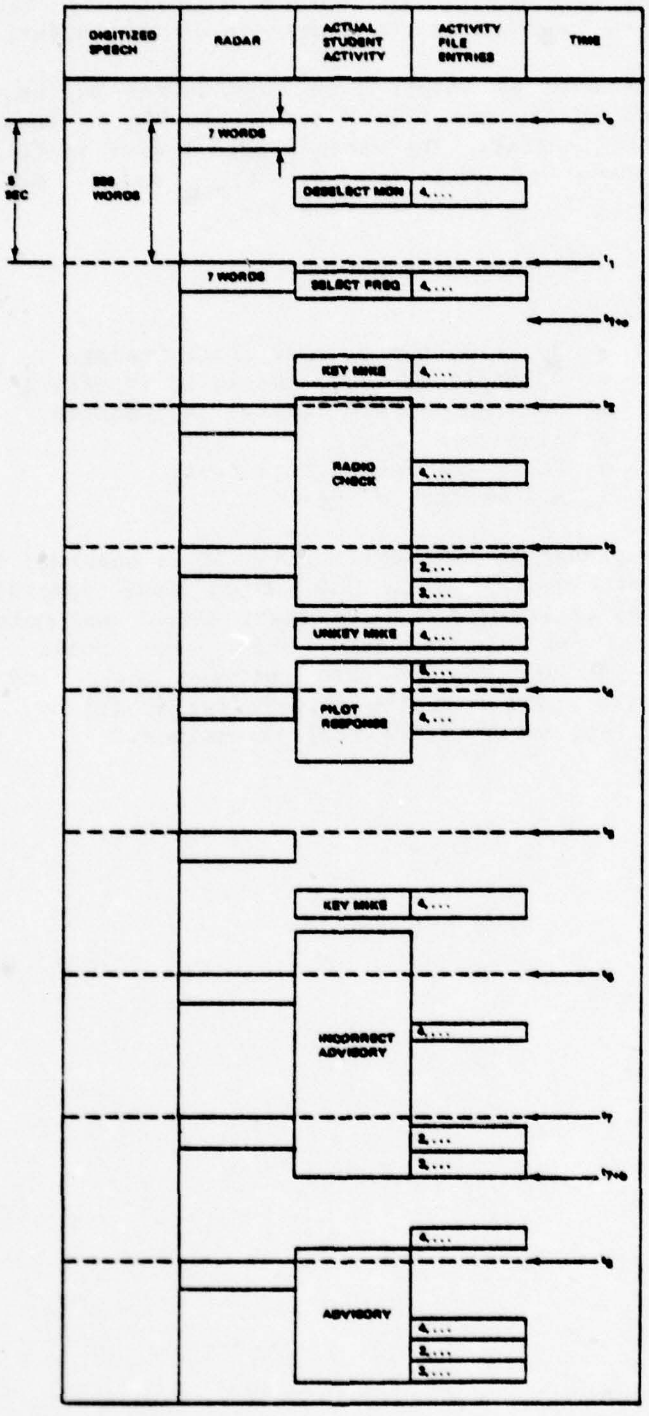


Figure 13. Replay Processing Example

t_6: When output from the t_6 buffer begins, SPBUF begins filling the other buffer with t_7 to t_{7+b} data. Nulls are inserted in the remainder of the buffer.

t_7: As soon as output from this buffer begins, busy is cleared to notify the speech output device to stop after output of this buffer. The other speech buffer is filled with nulls in place of the t_7 to t_{7+b} data, and the remaining t_{7+b} to t_8 data are read in.

t_{7+b}: Audible speech output stops

t_8: Radar:

- Receives the t_8 user clock message
- Determines that a freeze is in effect
- Does not update display information
- Stops user clock
- Sets time back 1 tick to t_7
- Allows PMS to speak

When the error explanation is complete, and it is observed that there is not another error within t_7 to t_8 (in which case special processing is necessary), replay is resumed. Both speech output and radar are started at t_7, using the saved speech data which has nulls between t_7 and t_{7+b}, and the actual speech data between t_{7+b} and t_8. Program descriptions of the routines which provide replay follow. Figure 14 indicates the interrelationship of the replay routines.

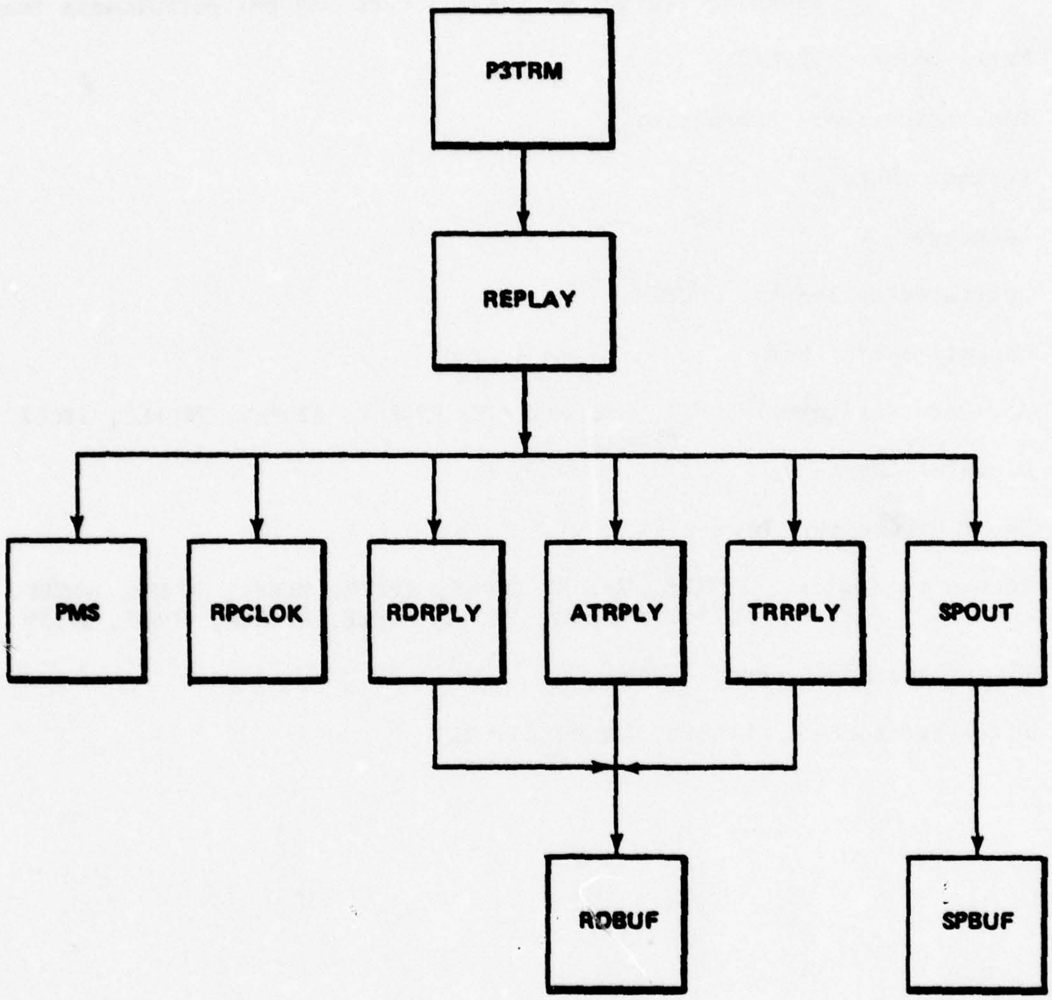


Figure 14. Relationship of Replay Routines Within the GCA-CTS Environment

REPLAY

Description: REPLAY initiates the processing which provides the various types of replays of phase 3 runs and the performance test.

Entry point: REPLAY

Classification: Subroutine

Period: None

Language: F

Activated/called by: P3TRM

Cancelled by: N/A

Activates/calls: IPBOUT1, PMS, RDRPLY, RPCLOK, ATRPLY, TRRPLY, SPOUT

Cancels: None

Input arguments: None

Common variables: RPTIME, RPTSP, RPRBF, RPPTR, RPSP1, RPSP2, RPSBF, NCDV, NCRPLY, NCRPAT, KYDIA, KYDIB, KYHOLD, RPABF, RPATR

Files created/changed: None

Files referenced: RPLSPH, RPLDSP, RPLACT

PROCEDURE REPLAY

DO REPLAY FILES EXIST?

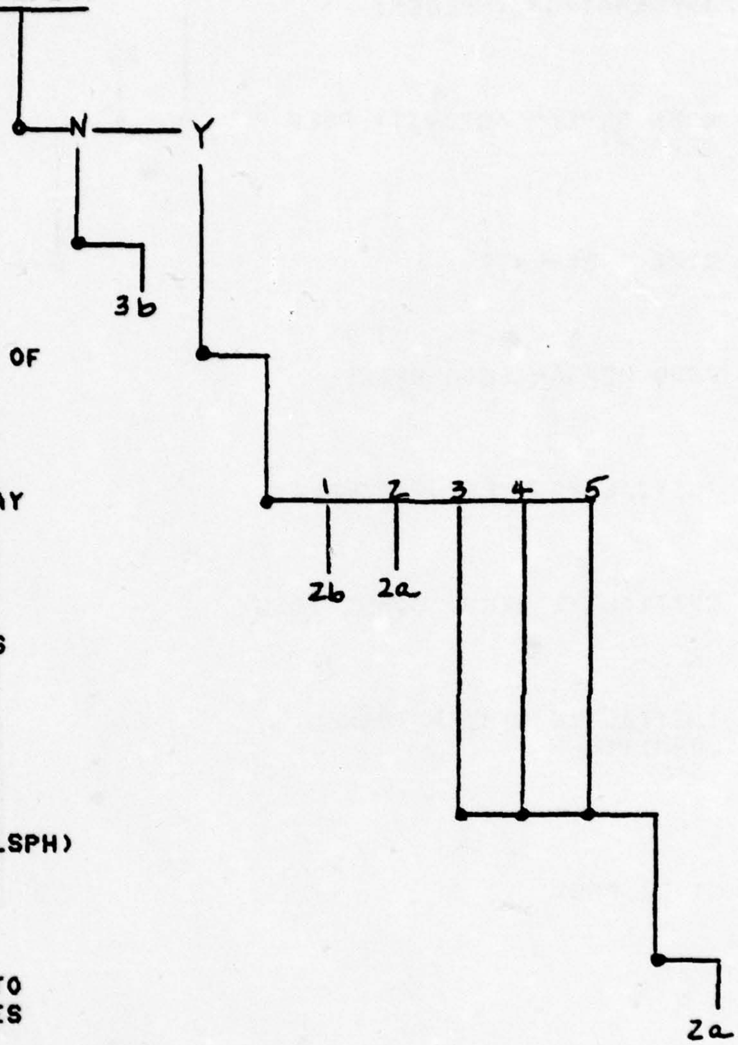
REPORT ERROR

INFORM DISPLAY OF TYPE OF
REPLAY

BRANCH ON TYPE OF REPLAY
REQUESTED:
1-SCORE ONLY
2-TRACK ONLY
3-REPLAY
4-REPLAY, REPEAT ERRORS
5-REPLAY, GIVE RULES

OPEN FILE OF DIGITIZED
SPEECH INFORMATION (RPLSPH)

NOTIFY STUDENT PANEL
INTERRUPT SERVICE NOT TO
MODIFY PANELS ON ENTRIES
(KYHOLD)



REPLAY FLOWCHART (SHEET 1 OF 3)

OPEN REPLAY FILE OF RADAR
INFORMATION (RPLDSP)

OPEN REPLAY ACTIVITY FILE
(RPACT)

TYPE 1 REPLAY?

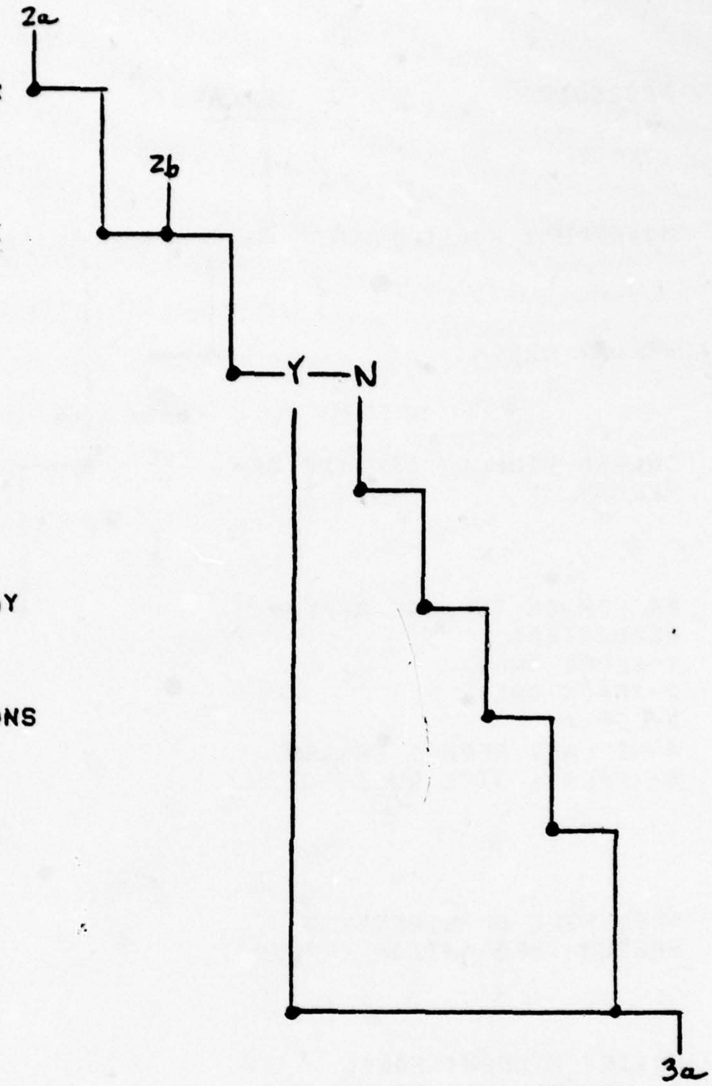
READ HEADER FROM RPACT

INITIALIZE TYPE OF DISPLAY

INITIALIZE SERVO CONDITIONS

INITIALIZE STUDENT PANEL
CONDITIONS

START PMS



REPLAY FLOWCHART (SHEET 2 OF 3)

BRANCH ON TYPE OF REPLAY

START TRACK REPLAY (TRRPLY)

START SPEECH REPLAY (SPOUT)

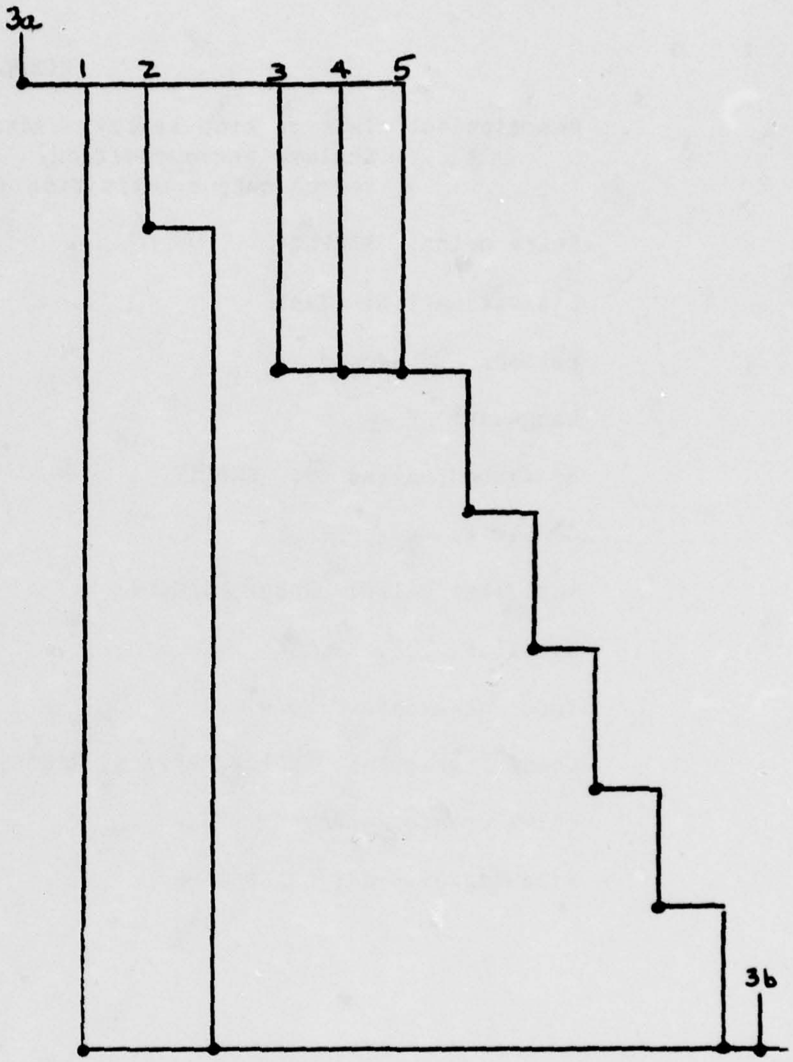
START RADAR REPLAY (RDRPLY)

START ACTIVITY REPLAY
(ATRPLTY)

INITIALIZE REPLAY CLOCKS

START REPLAY USER CLOCK
(RPCLOK)

RETURN



REPLAY FLOWCHART (SHEET 3 OF 3)

RDRPLY

Description: This routine replays radar outputs as they occurred and also replays servo position. It synchronizes its activities with speech output activities as required.

Entry point: RDRPLY

Classification: Task

Period: .5 second

Language: F

Activated/called by: REPLAY

Cancelled by: RDRPLY

Activates/calls: RDBUF, RPCLOK

Cancels: PMS, RPCLOK

Input arguments: None

Common variables: RPTSP, RPTIME, RPRBF, RPPTR, RPINIT, NCRPLY

Files created/changed: None

Files referenced: RPLDSP

PROCEDURE RDRPLY

SET SYNCHRONIZATION INDICATOR RPINIT TO 1

READ BUFFER 1 OF RADAR REPLAY DATA

READ BUFFER 2 OF RADAR REPLAY DATA

CALL REC, WAIT FOR RPCLOK TICK

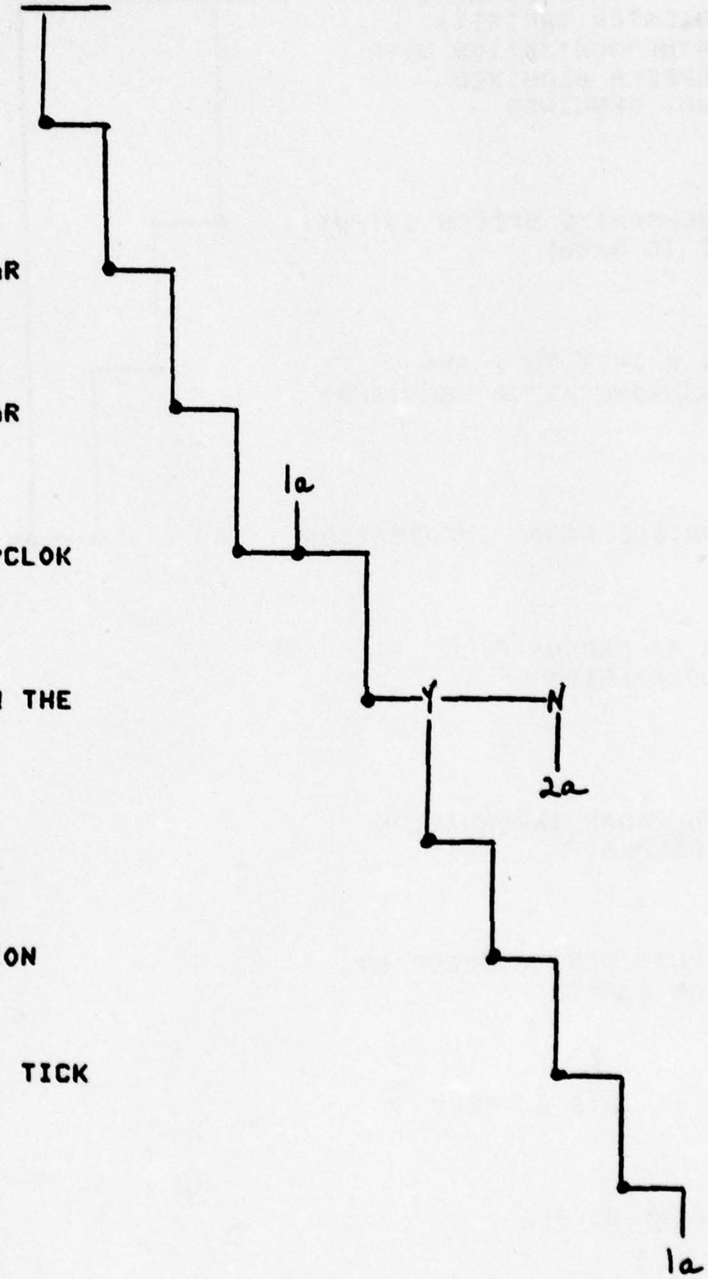
WAS THERE AN ERROR ON THE LAST TICK?

REMOVE USER CLOCK

START ERROR EXPLANATION

SET CLOCK TIME BACK 1 TICK

SET SYNCHRONIZATION INDICATOR TO 1



RDRPLY FLOWCHART (SHEET 1 OF 3)

BRANCH ON SYNCHRONIZATION
INDICATOR (RPINIT)
1-SYNCHRONIZATION WITH
SPEECH REQUIRED
2-NOT REQUIRED

SYNCHRONIZE SPEECH OUTPUT:
XMT TO SPOUT

SET RPINIT TO 2 (NO
SYNCHRONIZATION REQUIRED)

RETRIEVE RADAR INFORMATION

WAS AN END OF FILE
ENCOUNTERED?

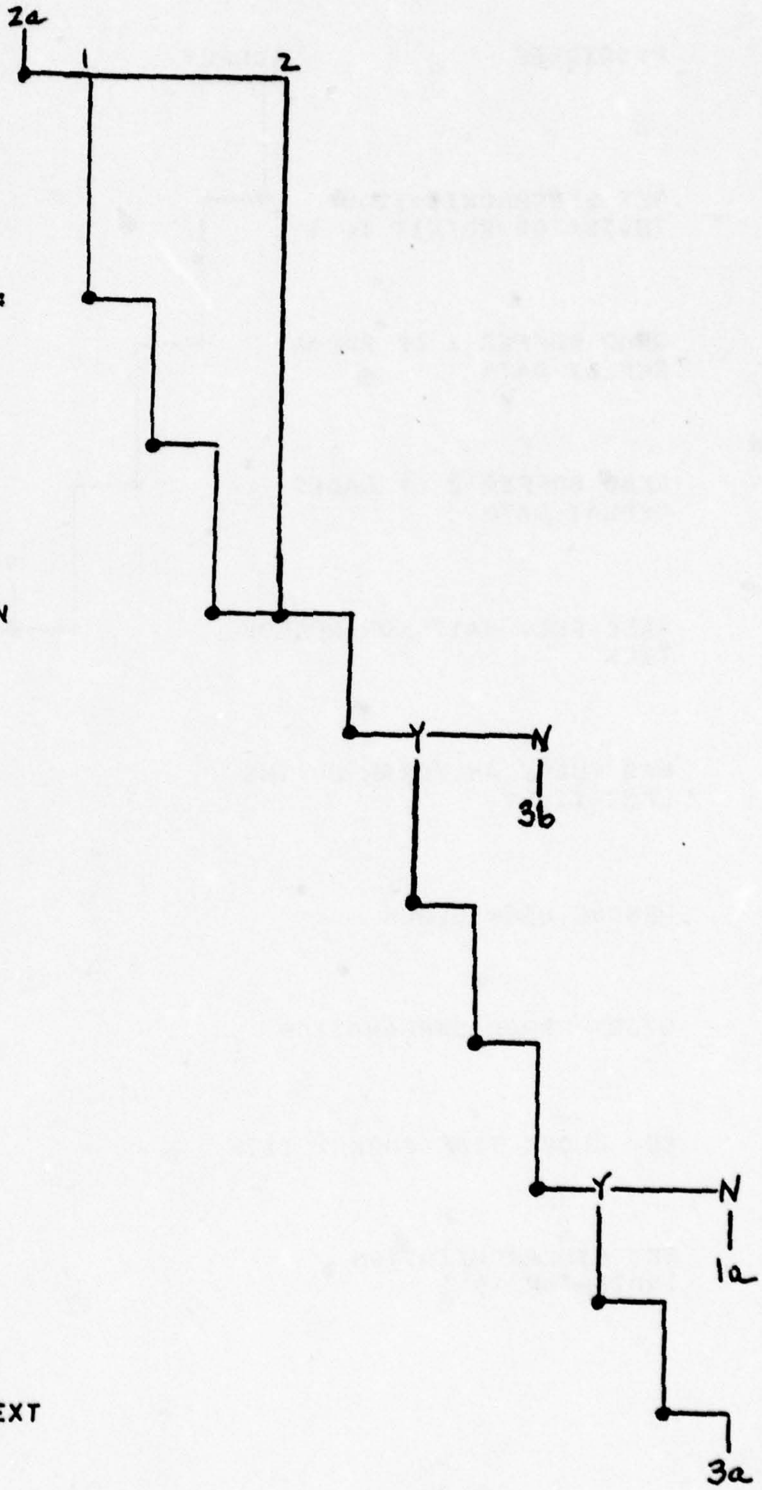
SEND RADAR INFORMATION
TO DISPLAY

COMPUTE NEXT ADDRESS IN
RADAR BUFFER

END OF THIS BUFFER?

RELEASE BUFFER

SET ADDRESS TO START OF NEXT
BUFFER



RDRPLY FLOWCHART (SHEET 2 OF 3)

3a

START RDBUF TO FILL RELEASED BUFFER

1a

3b

REMOVE USER CLOCK

INFORM CALLER BY WAKEUP EVPHZ

KILL PMS

CLOSE REPLAY FILES

RELEASE HOLD ON STUDENT PANEL

DIE

RDRPLY FLOWCHART (SHEET 3 OF 3)

TRRPLY

Description: TRRPLY produces a track history replay of the run.

Entry point: TRRPLY

Classification: Task

Period: None

Language: F

Activated/called by: REPLAY

Cancelled by: TRRPLY

Activates/calls: IPBOUT1, RDBUF

Cancels: None

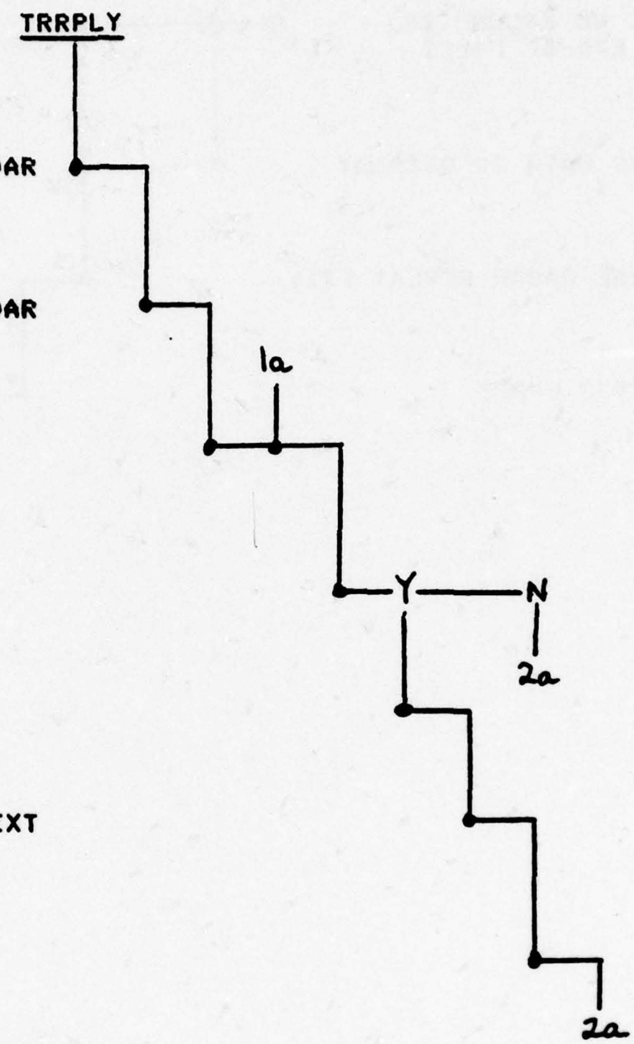
Input arguments: NSKP - Number of records to skip between displayed aircraft positions

Common variables: NCRPLY, RPRBF, RPPTR

Files created/changed: None

Files referenced: RPLDSP

PROCEDURE



ACTIVATE RDBUF TO FILL RADAR BUFFER 1

ACTIVATE RDBUF TO FILL RADAR BUFFER 2

COMPUTE NEXT ADDRESS IN RADAR BUFFER

END OF THIS BUFFER?

RELEASE BUFFER

SET ADDRESS TO START OF NEXT BUFFER

START RDBUF TO FILL THIS BUFFER

TRRPLY FLOWCHART (SHEET 1 OF 2)

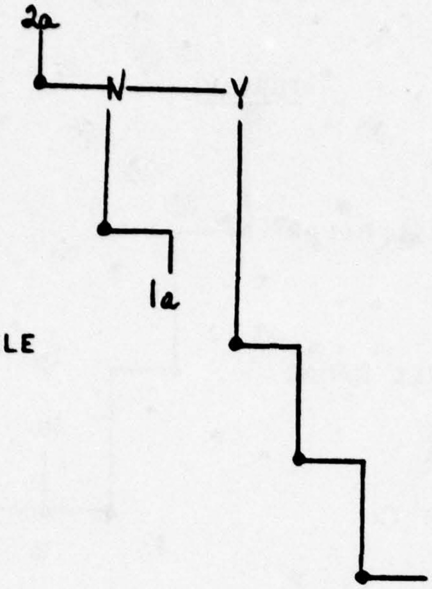
DID WE ENCOUNTER
AN END OF FILE?

SEND DATA TO DISPLAY

CLOSE RADAR REPLAY FILE

WAKEUP EVPHZ

DIE



TRRPLY FLOWCHART (SHEET 2 OF 2)

ATRPLY

Description: ATRPLY synchronizes the replay of student panel inputs, servo inputs, and synthesized speech outputs with the rest of replay.

Entry point: ATRPLY

Classification: Task

Period: None

Language: F

Activated/called by: REPLAY

Cancelled by: ATRPLY

Activates/calls: RDBUF, IPBOUT1

Cancels: None

Input arguments: None

Common variables: RPALK1, RPALK2, RPBLK1, RPBLK2, RPTAT, RPATR, RPTIME, RPABF, NCRPAT, BXACT

Files created/changed: None

Files referenced: RPLACT

PROCEDURE ATRPLY

START RDBUF TO FILL
ACTIVITY BUFFER A

START RDBUF TO FILL ACTIVITY
BUFFER B

SET CURRENT BUFFER POINTER
TO BUFFER A

COMPUTE POINTER INTO
ACTIVITY BUFFER

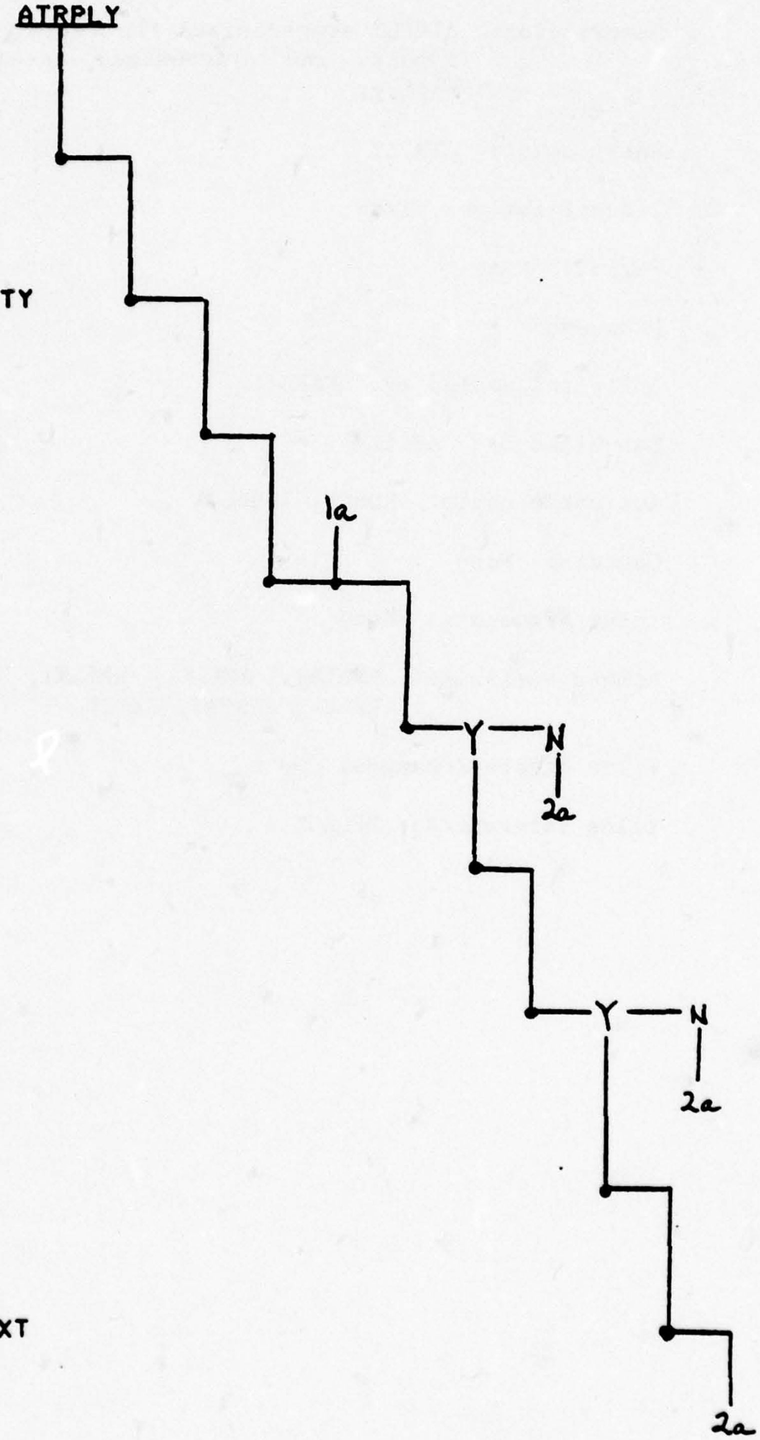
IS THIS THE END OF THE
BUFFER?

RELEASE LOCK 1 ON THIS
BUFFER

HAS PMS ALSO FINISHED WITH
THIS BUFFER (LOCK 2
RELEASED?)

START RDBUF TO FILL THIS
BUFFER

SET POINTER TO START OF NEXT
BUFFER



ATRPLY FLOWCHART (SHEET 1 OF 2)

WAS END OF FILE ENCOUNTERED?

CLOSE ACTIVITY FILE

DIE

WAS THIS ACTIVITY OF TYPE 4
OR 5?

SET TIME OF ACTIVITY

IS TIME OF ACTIVITY <=
CURRENT TIME?

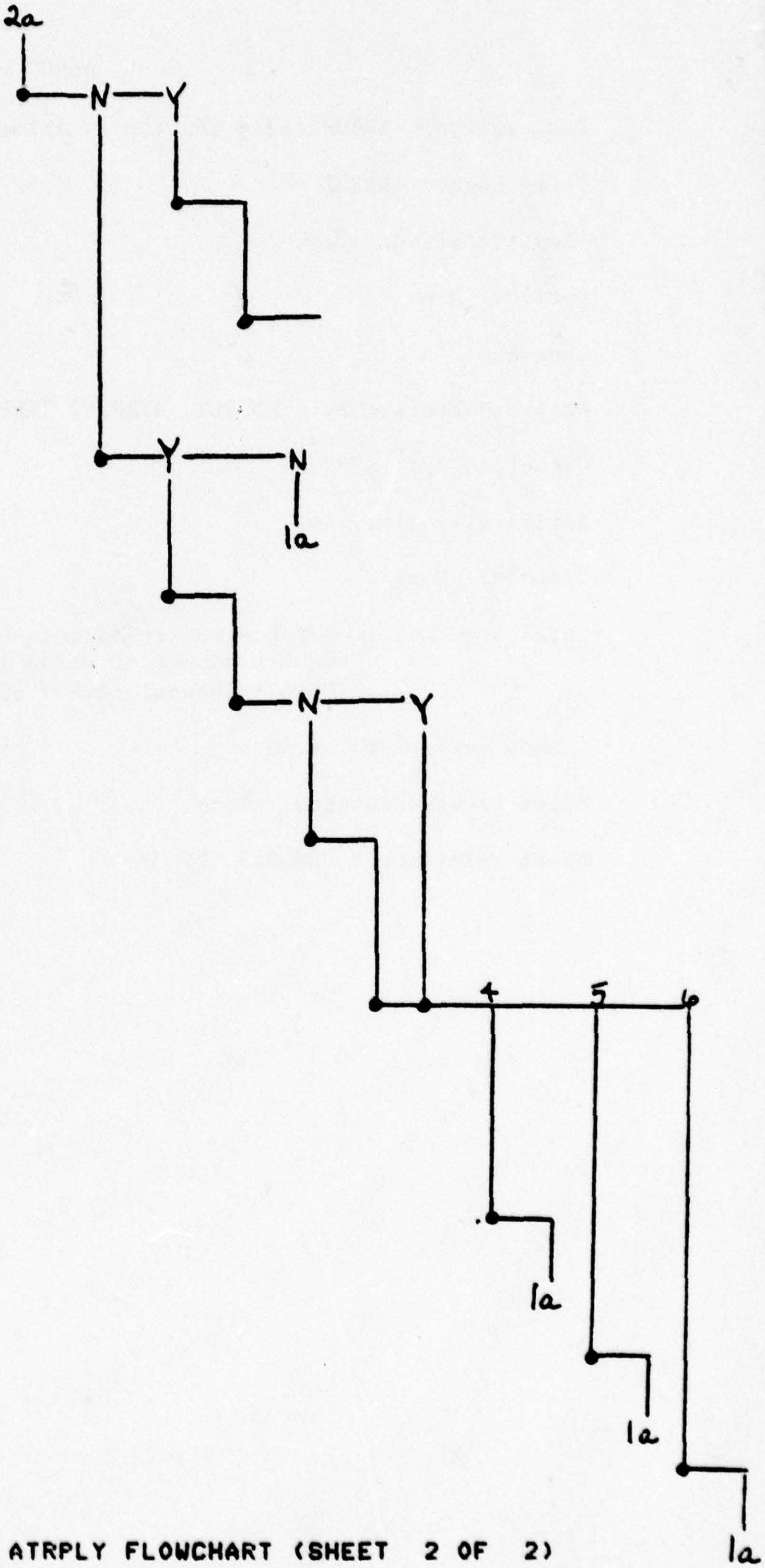
CALL REC, WAITING FOR USER
CLOCK TICK

BRANCH ON TYPE OF ACTIVITY:
4-STUDENT PANEL INPUTS
5-NET SERVO CHANGES
6-AUTOMATED SPEECH
  OUTPUTS

CHANGE STATE OF STUDENT OR
INSTRUCTOR PANEL

CHANGE STATE OF SERVO

OUTPUT VERBAL ADVISORY



ATRPLY FLOWCHART (SHEET 2 OF 2)

RDBUF

Description: RDBUF fills the replay information buffers.

Entry point: RDBUF

Classification: Task

Period: None

Language: A

Activated/called by: RDRPLY, ATRPLY, TRRPLY

Cancelled by: RDBUF

Activates/calls: None

Cancels: None

Input arguments: BUFAD - Buffer address of buffer to be filled
NWDS - Number of words in buffer
CHNL - Channel number of file

Common variables: None

Files created/changed: None

Files referenced: RPLDAT, RPLDSP

PROCEDURE RDBUF

COPY BUFFER ADDRESS ONTO STACK

READ ONE BUFFER OF DATA FROM REPLAY FILE

EOF ENCOUNTERED? Y N

PACK EOF MARKER IN BUFFER

DIE

RDBUF FLOWCHART (SHEET 1 OF 1)

RPCLOK

Description: The replay user clock maintains a 100 msec. clock and a .5 second clock. The ticking of the .5 second clock causes an IXMT which activates the radar display replay task RDRPLY. It also starts the activity processing module at the appropriate time.

Entry point: RPCLOK

Classification: User clock

Period: 100 msec

Language: A

Activated/called by: REPLAY, RDRPLY

Cancelled by: RDRPLY

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: RPTIME, BXRPL, RP100, BXACT

Files created/changed: None

Files referenced: None

PROCEDURE RPCLOK

ENTER ON 100 MSEC USER CLOCK INTERRUPT

HAS .5 SECOND ELAPSED?

INCREMENT RP100

SET RP100 TO 0

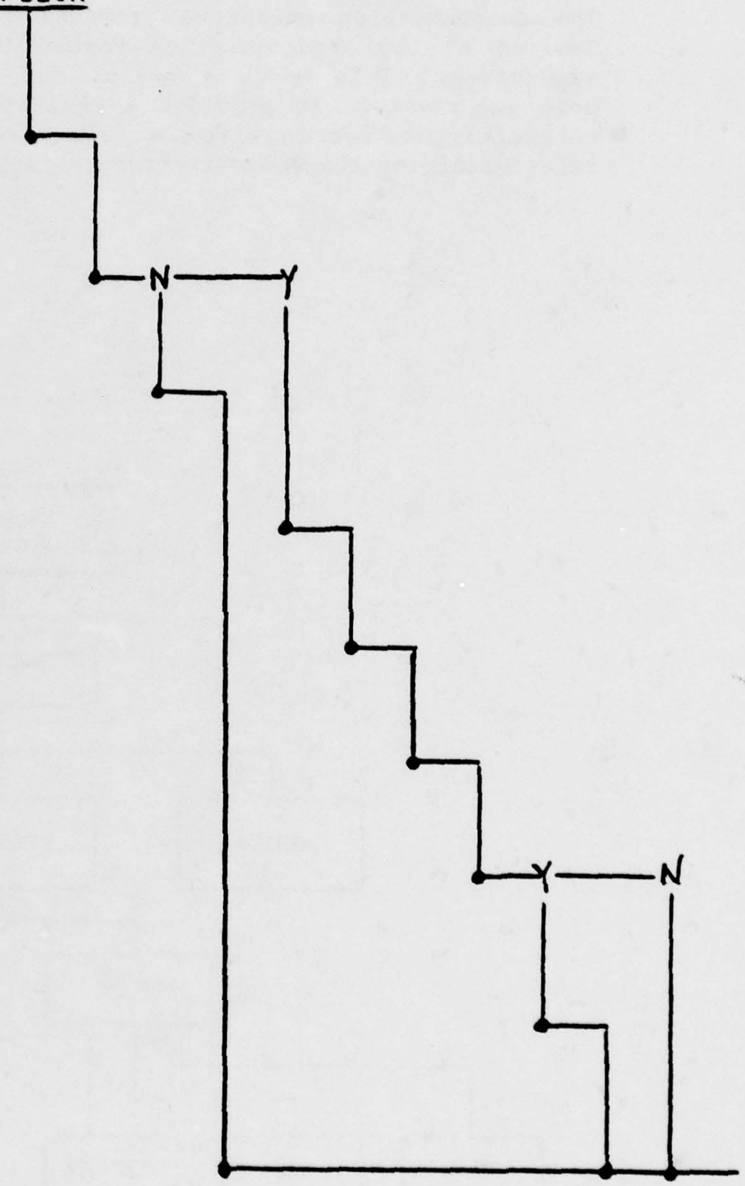
INCREMENT RPTIME

.IXMT TO RDRPL

IS IT TIME FOR AN ACTIVITY TO BE PROCESSED?

IXMT TO ATRPL

EXIT



RPCLOK FLOWCHART (SHEET 1 OF 1)

Demonstration

The demonstration executive provides problems for the model controller instead of the student. The result is that the system itself conducts approaches. This mode is useful for demonstrating system capabilities. More importantly, it provides a realistic radar control environment for a natural transition to alignment checking procedures. Figure 15 shows the relationship of the demonstration routines within the GCA-CTS environment.

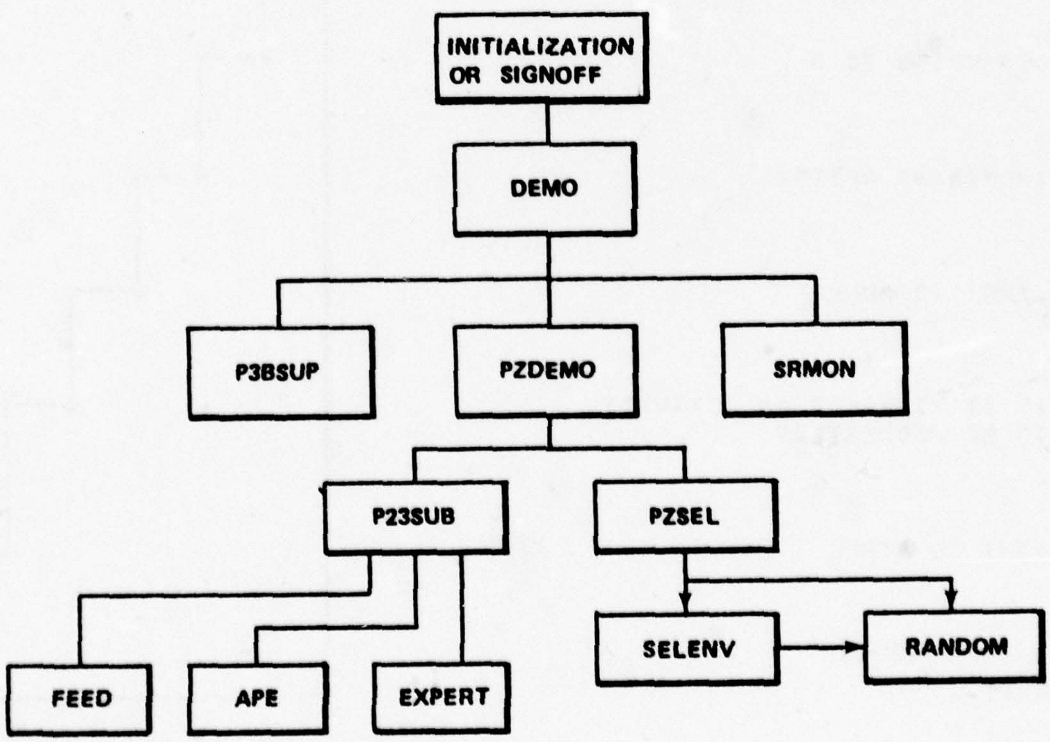


Figure 15. Relationship of Demonstration Routines in the GCA-CTS Environment

DEMO

Description: DEMO conducts all types of approaches during system idle time. It provides for a smooth transition to alignment checking when the student signs on. It also serves to demonstrate system capabilities on request.

Entry point: DEMO

Classification: Task

Period: None

Language: F

Activated/called by: INIT1, KBRD, SGNOFF

Cancelled by: SGNON

Activates/calls: P3BSUP, PZDEMO, SRMON, IPBOUT, GLIB, SMENU, IMENU

Cancels: None

Input arguments: None

Common variables: GZPHZ, NCPHZ, RANDOM, SMENU, IMENU, GZSEED, NCAO

Files created/changed: Demo setup

Files referenced: None

PROCEDURE DEMO

SET PHASE TO 6

MODIFY STUDENT MENU

MODIFY INSTRUCTOR MENU

OPEN DEMO PROBLEM FILE

READ COMMENT(S), DISPLAY AT INSTRUCTOR STATION

READ FIRST CARD OF ENVIRONMENTAL SIMULATION CARD SET

CALL P3BSUP TO ENCODE INFORMATION

UPDATE RANDOM NUMBER SEED VALUE IN DEMO PROBLEM FILE (GZSEED)

CLOSE DEMO PROBLEM FILE

NOTIFY CPU2 TO WRITE "DEMO IN PROGRESS" ON CRT

2a

DEMO FLOWCHART (SHEET 1 OF 2)

2a

SET NOMINAL SERVO POSITION

START SRMON

START PZDEMO

DIE

DEMO FLOWCHART (SHEET 2 OF 2)

PZDEMO

Description: PZDEMO oversees demonstration run presentations.

Entry point: PZDEMO

Classification: Task

Period: None

Language: F

Activated/called by: DEMO

Cancelled by: PZDEMO

Activates/calls: PZSEL, P23SUB, TZEC

Cancels: None

Input arguments: None

Common variables: (Menu selections)

Files created/changed: None

Files referenced: None

PROCEDURE PZDEMO

CALL PZSEL

PROCESS SPECIAL DEMO REQUESTS

CALL P23SUB

ASK MODEL CONTROLLER TO PERFORM THE DEMONSTRATION

WAIT FOR EVPHZ

EVPHZ HAPPENED

HAS A TERMINATE REQUEST BEEN RECEIVED?

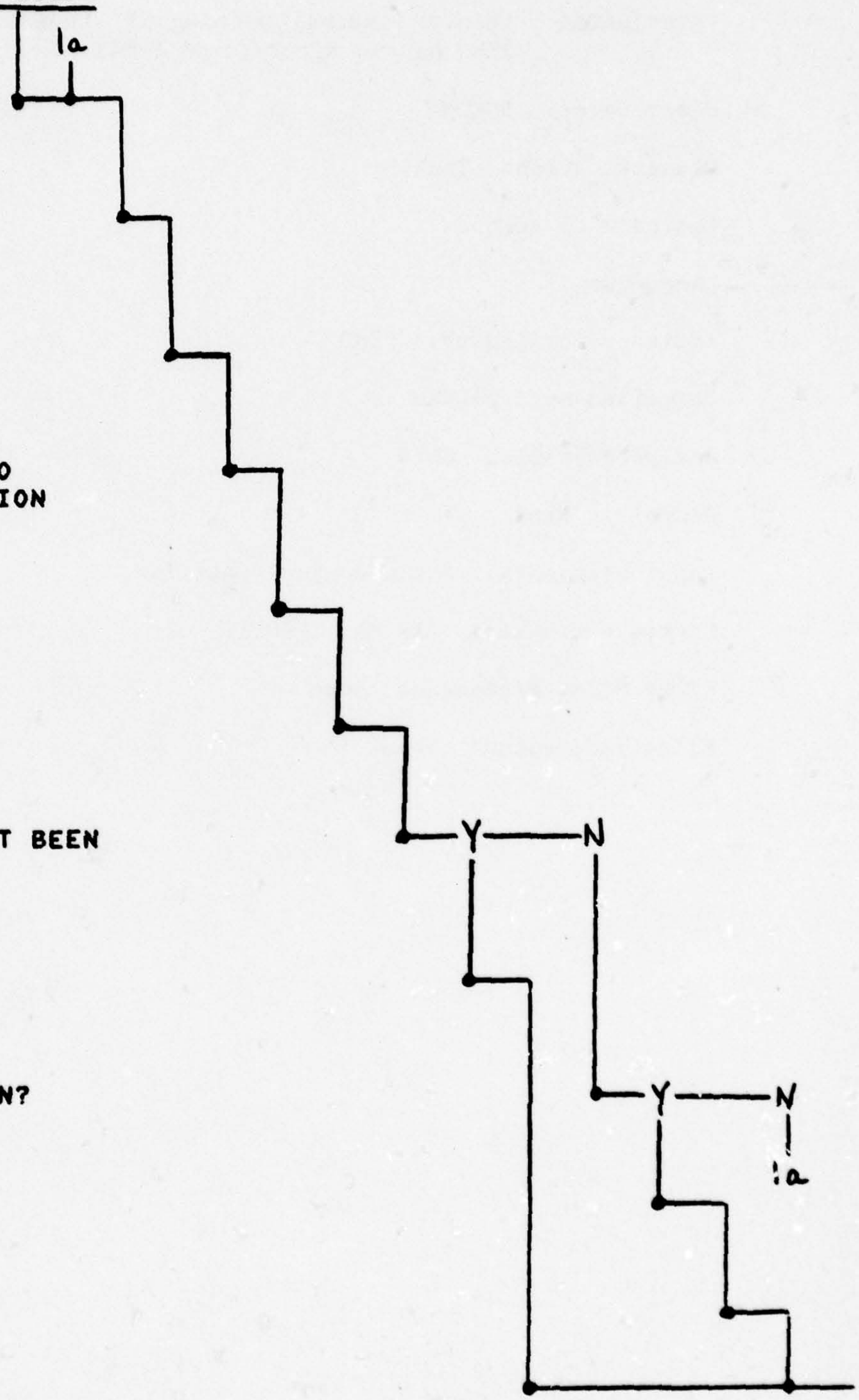
NOTIFY CPU2 TO QUIT

HAS A STUDENT SIGNED ON?

CANCEL SRMON

START TZEC

DIE



PZDEMO FLOWCHART (SHEET 1 OF 1)

SRMON

Description: Issues a verbal warning if student activates the servo while DEMO has an aircraft on final.

Entry point: SRMON

Classification: Task

Period: .5 second

Language: F

Activated/called by: DEMO

Cancelled by: PZDEMO

Activates/calls: GLIB

Cancels: None

Input arguments: Servo nominal position

Common variables: BXSRMN, IPBOUT1

Files created/changed: None

Files referenced: None

PROCEDURE SRMON

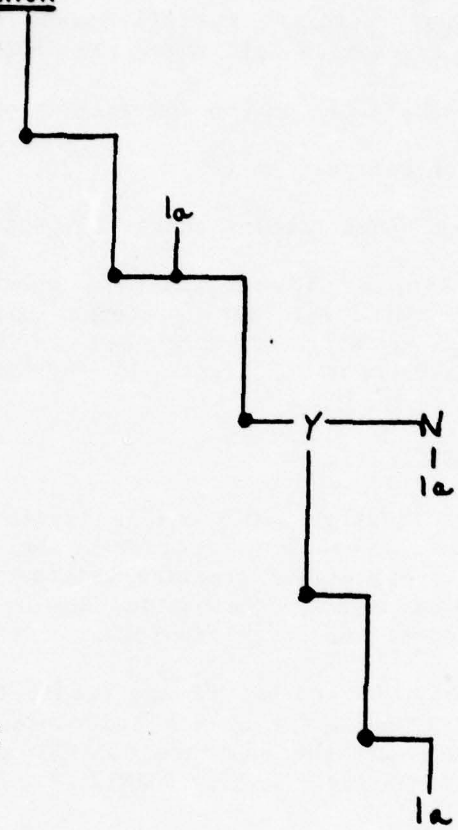
SAVE CURRENT SERVO POSITION IN TEMPORARY

CALL REC FOR MESSAGE FROM USER CLOCK (BXSRMN)

IS PRESENT SERVO POSITION DIFFERENT FROM NOMINAL POSITION?

RESET SERVO POSITION TO NOMINAL VALUE

WARN USER NOT TO SERVO WHILE ANOTHER CONTROLLER IS USING THE RADAR



SRMON FLOWCHART (SHEET 1 OF 1)

Speech Related Software

SPEECH is the controller for all speech related software handled by CPU2. Three major categories fall under the SPEECH's jurisdiction:

1. Voice data collection and validation (VDC)

2. Speech recognition (SR)

3. Speech understanding subsystem (SUS)

VDC validation and SUS utilize the speech recognition modules. SUS is dependent on voice reference patterns collected during VDC. The design of each of these parts is discussed in further detail in the following discussions. Refer to figure 16 for an overview of the speech related software.

Voice Data Collection

Voice data collection (VDC) and validation is inherent to phase 1 instruction. While the student is taught the correct radio terminology, he is encouraged to repeat and practice voicing the phrases. During this process, voice input feature patterns may be easily obtained without requesting unnatural repetition of terminology.

The routines VDCON and VDCOFF are required to prepare for VDC and to terminate VDC. Channels are opened and closed to the voice data files. After VDC is turned on, the routines COLLECT and FORMIT may be used to COLLECT input feature patterns and to FORMIT into voice reference patterns.

Input Collection. COLLECT does not issue prompts automatically but does re-prompt if a faulty input feature pattern is detected: incorrect pauses, incorrect concatenations or a long drawn-out phrase. It does so by counting phrases input and checking for buffer overflow. The buffer is examined for leftover phrases input after the correct number is collected.

The features are input via the Threshold 500 driver (VIPDR) to be processed by input feature pattern formulation routine VIFP.

Input Feature Patterns (IFPs). Every 2 milliseconds (approximately) the Threshold 500 preprocessor generates an interrupt and provides two 16-bit words via the interface logic. Each of these 32 bits corresponds to a feature. A bit will be set (i.e., equal to a one) if, and only if, the corresponding feature was present in the voice sample.

All of these data are stored in memory by an interrupt service routine (VIPDR). One of the features (LP_4) indicates a long pause. When this feature is set, the system assumes that the phrase is complete and stops accepting data from the preprocessor.

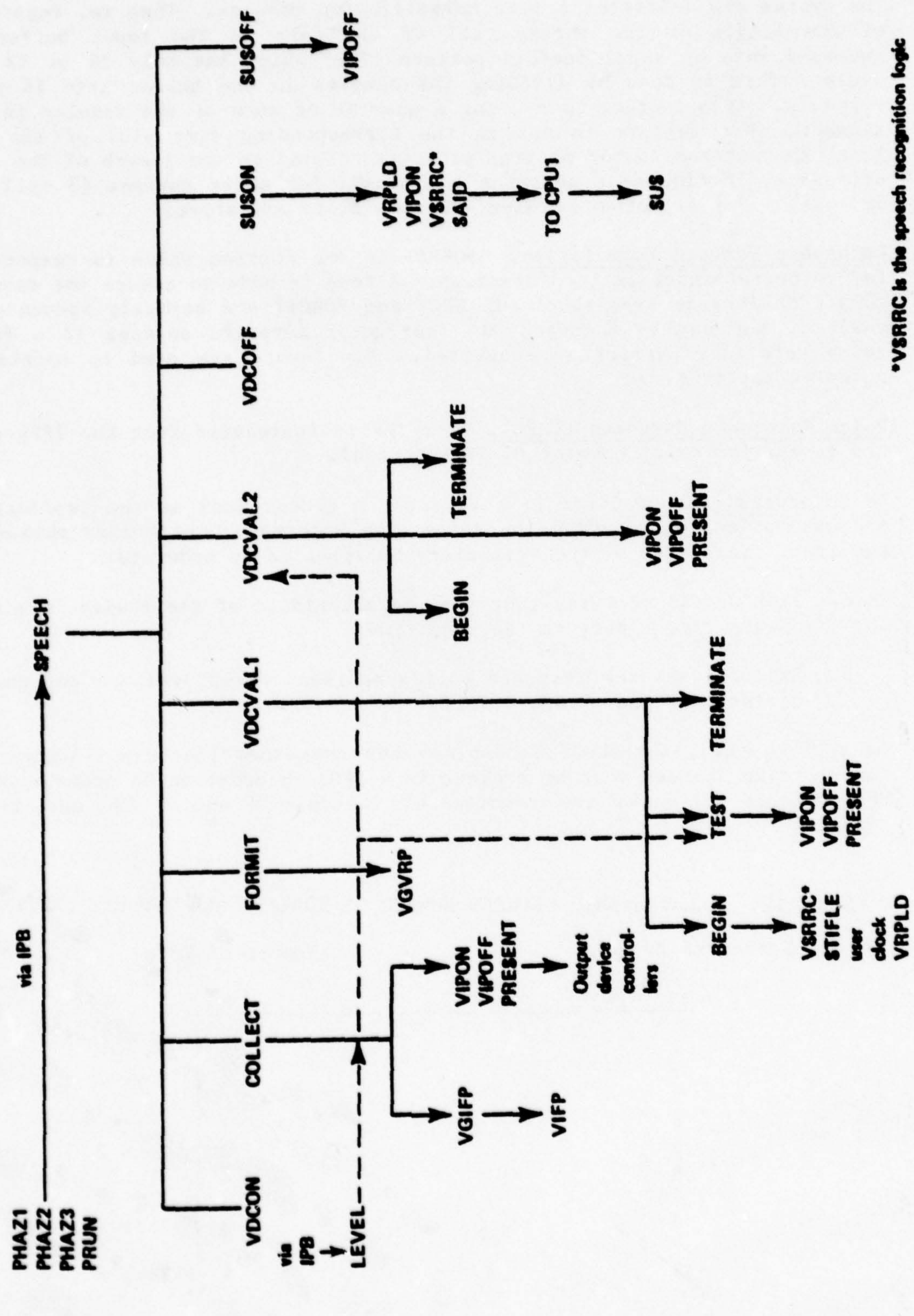


Figure 16. Speech Related Routines

The system now initiates a time normalization process. That is, regardless of the length of the phrase, all of the data in the input buffer are squeezed into an input feature pattern (IFP) which has only 16 or 32 time slots. This is done by dividing the samples in the buffer into 16 or 32 segments. If a feature is set for a quarter or more of the samples in each segment, that feature is set in the corresponding time slot of the IFP. Since the optimum number of time slots is related to the length of the input utterance, 16 time slot patterns are stored for short phrases (3 syllables or less). For all other phrases, 32 time slots are stored.

Reference Pattern Formulation. FORMIT is the routine which is responsible for voice reference pattern creation. A test is made to assure the required number of IFPs is available. COLLECT and FORMIT are normally requested by phase 1, but may be accessed via instructor keyboard entries if a faulty voice reference pattern is suspected. New inputs are used to update the outdated patterns.

Voice Reference Patterns (VRPs). The VRP is formulated from the IFPs using the repetition count (number of IFPs stored).

To understand the problems in this area, a closer look at the implications and assumptions surrounding the repetitive voicing of vocabulary phrases is required. Each item of the vocabulary is repeated in order to:

a. Pick up all features that are characteristic of the student's voicing of a phrase — but, at the same time,

b. Exclude features that are only sometimes present and are not characteristic of the phrase itself.

To achieve this, Threshold Technology has suggested that for M repeats, a feature/time element must be present in N IFPs in order to be present in the VRP. Table 12 shows the relationship between M and N for one to ten repeats.

TABLE 12. RELATIONSHIP BETWEEN NUMBER OF REPEATS AND IFP BIT SETTINGS

Number of Repeats M	Number of IFPs N	
1	1	Zone 1
2	1	
3	1	
4	2	Zone 2
5	2	
6	2	
7	3	Zone 3
8	3	
9	3	
10	4	Zone 4

Generally speaking, the repetition count can be considered in zones: 1-3 repetitions in the first zone, 4-6 in the second, 7-9 in the third, and 10 in the fourth. The second objective is furthered by moving from zone to zone. The first objective is furthered by increasing the repetition count within a zone.

Notice that the first three repetitions will put all features into the VRP, thus no contribution is made toward the second objective previously cited.

With the fourth repetition, the number of features put into the VRP is reduced considerably, but the number starts to build up again until the seventh repetition, and so on. The number of features set in the VRP is therefore related — in a very nonlinear way — to the number of repetitions.

The score that will be calculated during the speech recognition process by correlating an IFP with each VRP is a function of the number of features set in the VRP (and IFP). So, to compare scores, it is necessary to ensure that all VRPs have been formed with the same probability of success.

Past tests have produced the following correlation between score and repetition count.

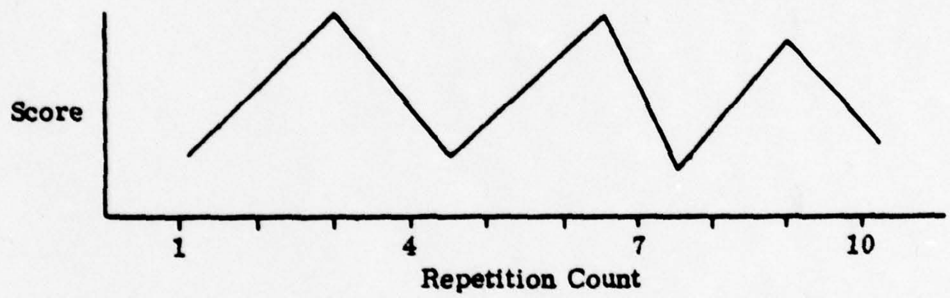


Figure 17. Score as a Function of Repetition Count.

Therefore, the following conclusions resulted:

a. No phrase should be trained less than four times

b. Phrases which are easily confused (e.g., above glidepath and below glidepath) should be trained an equal number of times.

In implementation, then, four IFPs are collected for the phrases which are not likely to be mistaken for another phrase. For other phrases which are likely to produce confusion, ten IFPs are collected before a voice reference pattern is formulated.

Validation. The purpose of this mode of VDC program operation is to further instruct a student and/or allow him practice with the radio terminology. Familiarization with the speech recognition system is a beneficial side effect.

The validation mode actually consists of two submodes, VDCVAL1 and VDCVAL2. Using VDCVAL2, the system will not prompt the student; rather, the student prompts the system. That is, the system will attempt to echo the phrases which it recognizes the student to be speaking. This presentation occurs one second after the student is silent.

In the VDCVAL1 mode, the program will prompt the student in the same way as in the training mode, utilizing the latest presentation device. Recognition accuracy is recorded in the voice data file. When the requested recognition accuracy has been achieved for three consecutive prompts, the validation mode is terminated.

Subroutines BEGIN and TERMINATE apply to both validation modes. They perform the functions the names imply. PRESENT serves both VDCVAL1 and VDCVAL2 as a prompting and mimicking routine, respectively.

SPEECH

Description: SPEECH initiates the speech related routines which accomplish voice pattern collection, voice pattern validation, recognition, and understanding.

Entry point: SPEECH

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN2 (by PHAZ*)

Cancelled by: N/A

Activates/calls: VDCON, COLLECT, FORMIT, VDCVAL1, VDCVAL2, VDCOFF, SUSON, SUSOFF

Cancels: None

Input arguments: SECOM - Command code (see table below)
SELST(1) - % validation for command code 4; phrase #
SELST(2), SELST(3), etc.-phrase #

Common variables: SPFLG, SPLST, SPDEV, EVPHZ

Files created/changed: None

Files referenced: None

COMMAND CODES

1-Start VDC	5-Validate, no prompt
2-Collect IFP	6-Stop VDC
3-Form VRP	7-Start SUS
4-Validate	8-Stop SUS

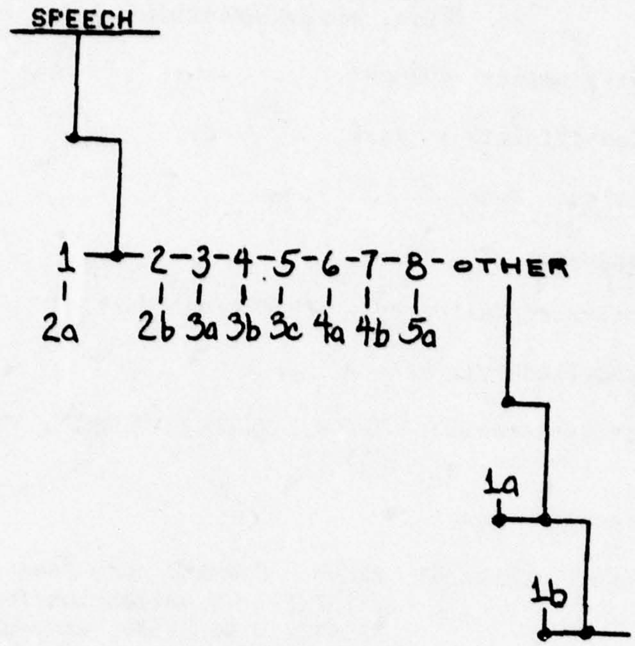
SET SPEECH DEVICE

BRANCH ON ARGUMENT

"SPEECH ERR."

WAKEUP PHASE EXECUTIVE

RETURN



SPEECH FLOWCHART (SHEET 1 OF 5)

IS SPEECH FLAG SET TO SUS?

"SUS IS ON ERR."

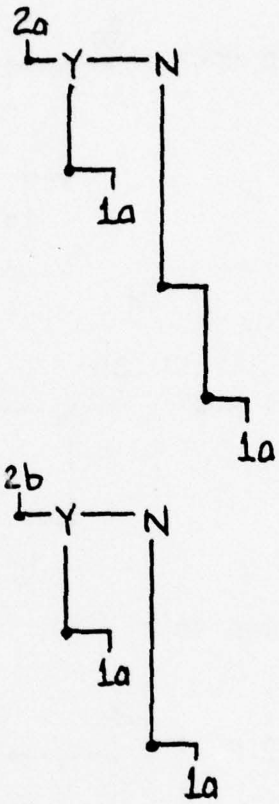
SET FLAG TO VDC.*

DO VDC INITIALIZATION

IS SPEECH FLAG SET TO VDC?

"SET VDC ERR."

COLLECT INDICATED IFPS



.SUS = SPEECH UNDERSTANDING SUBSYSTEM
*VDC = VOICE DATA COLLECTION (AND VALIDATION)

SPEECH FLOWCHART (SHEET 2 OF 5)

IS SPEECH FLAG SET TO VDC?

"SET VDC ERR."

FORM VRP

IS SPEECH FLAG SET TO VDC?

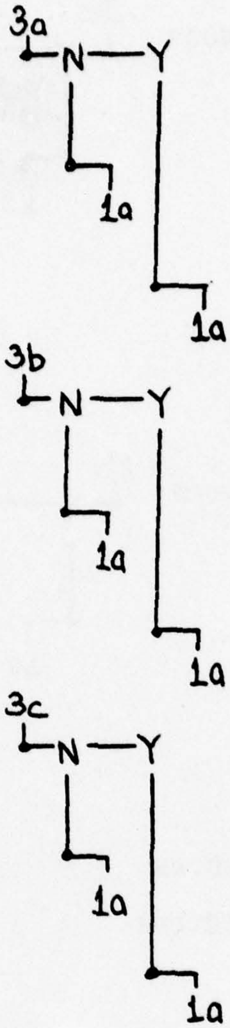
"SET VDC ERR."

DO VALIDATION WITH PROMPTS

IS SPEECH FLAG SET TO VDC?

"SET VDC ERR."

DO VALIDATION WITHOUT
PROMPTS



SPEECH FLOWCHART (SHEET 3 OF 5)

IS SPEECH FLAG SET TO VDC?

"SET VDC ERR."

RESET FLAG

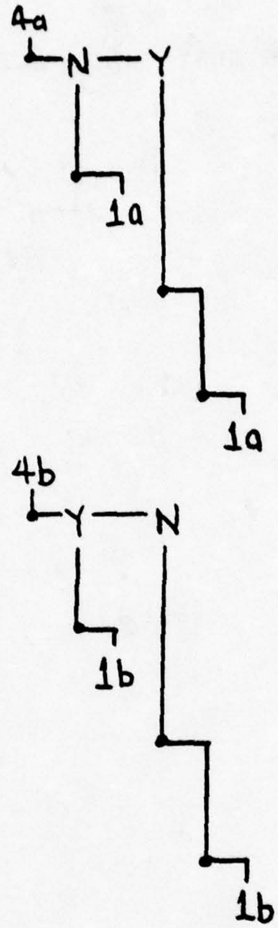
TERMINATE VDC

IS SPEECH FLAG SET TO VDC?

"VDC IS ON ERR."

SET FLAG TO SUS

DO SUS INITIALIZATION



SPEECH FLOWCHART (SHEET 4 OF 5)

IS SPEECH FLAG SET TO SUS? 5a N Y

"SET SUS ERR." 1b

RESET FLAG

TERMINATE SUS 1b

SPEECH FLOWCHART (SHEET 5 OF 5)

VDCON

Description: This routine opens all files necessary for VDC and validation and initializes all necessary trainee dependent arrays.

Entry point: VDCON

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: None

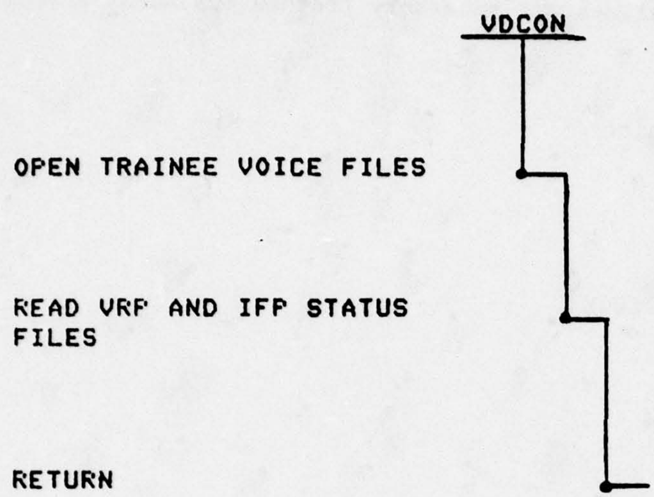
Cancels: None

Input arguments: None

Common variables: VCFRAZ, VCWGT, SPAT

Files created/changed: FNIFP, FNVRP

Files referenced: None



VDCON FLOWCHART (SHEET 1 OF 1)

COLLECT

Description: This routine collects input feature patterns from trainee voice inputs. Automatic prompts are provided only if a bad input is detected. The user must provide the initial prompt.

Entry point: COLLECT

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: VGIFP, PRESENT, VIPON, VIPOFF

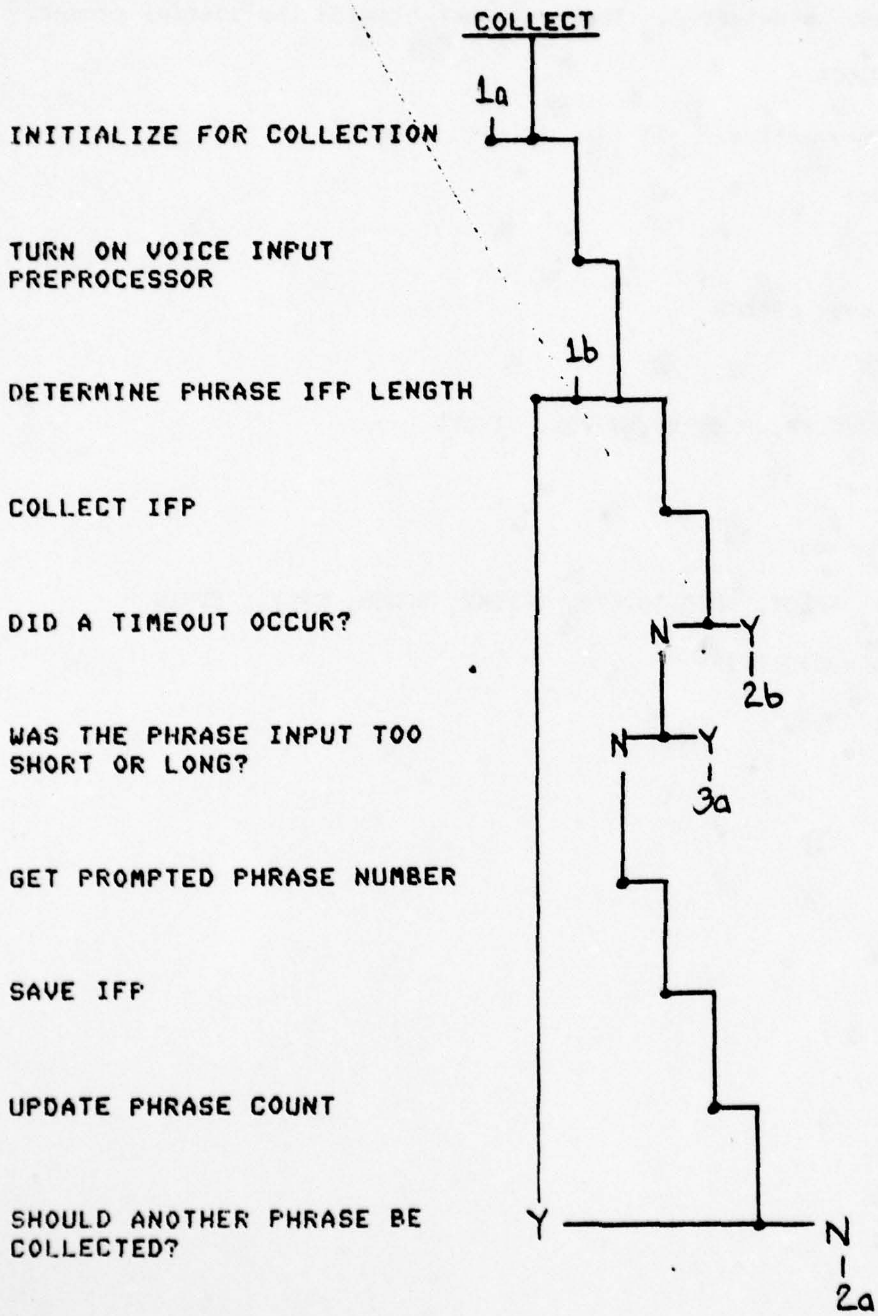
Cancels: VIP driver

Input arguments: None

Common variables: SPLST, SPID, SPIFP, VCFRAZ, SPNUM, SPLVL, EVVIN

Files created/changed: FNIFP

Files referenced: None



COLLECT FLOWCHART (SHEET 1 OF 3)

WAIT FOR SILENCE

IS PART OF A PHRASE STILL LEFT OVER?

SATISFACTORY VOICE INPUT LEVEL?

UPDATE IFP COLLECTION COUNTERS FOR NEW IFPS

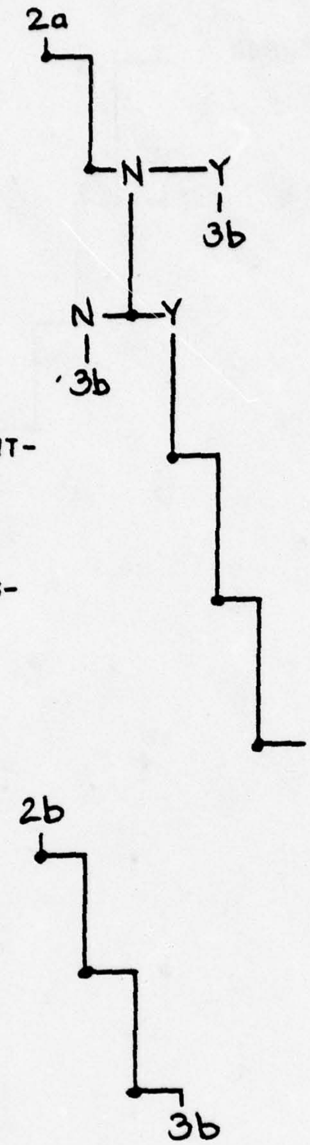
TURN VOICE INPUT PREPROCESSOR OFF

RETURN

TYPE TIMEOUT MESSAGE

TURN VIP OFF

PAUSE FOR STUDENT INPUT



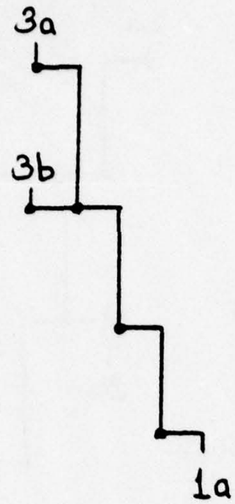
COLLECT FLOWCHART (SHEET 2 OF 3)

TYPE PHRASE LENGTH ERROR MESSAGE

DELETE COLLECTED DATA

TURN VIP OFF

PROMPT PHRASE(S) TO BE COLLECTED ON CRT



COLLECT FLOWCHART (SHEET 3 OF 3)

LEVEL

Description: This routine is activated when the IPB sends an end of student input message along with the input level to CPU2.

Entry point: LEVEL

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN2 event processor

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: SPLVL, EVVIN

Files created/changed: None

Files referenced: None

LEVEL

STORE INPUT LEVEL

SEND WAKEUP FOR STUDENT SILENCE

DIE

LEVEL FLOWCHART (SHEET 1 OF 1)

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

AD
A069036

VGIFP

Description: Creates input feature patterns (IFPs) for voiced inputs. If an input is not received within 20 seconds, a message is returned to the caller.

Entry point: VGIFP

Classification: Subroutine

Period: None

Language: F

Activated/called by: TRAIN

Cancelled by: N/A

Activates/calls: User clock service (VUCLK), VIFP

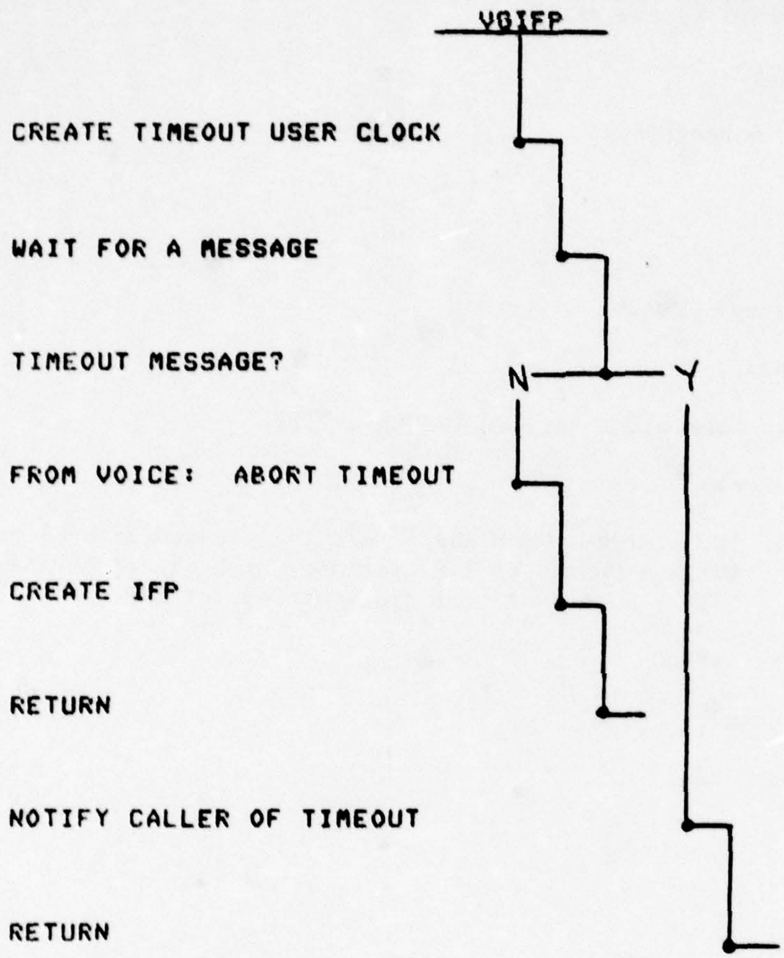
Cancels: User clock

Input arguments: IFP - Array which shall hold the created IFP (64 words)
IVTIM - Number of 2 ms periods which the IFP spans
IVTS - Number of time slots stored for VRP

Common variables: BXREC

Files created/changed: None

Files referenced: None



VGIFP FLOWCHART (SHEET 1 OF 1)

VIFP

Description: Creates an input feature pattern and input time at a location provided upon call.

Entry point: VIFP

Classification: Subroutine

Period: None

Language: A

Activated/called by: VGIFP

Cancelled by: N/A

Activates/calls: None

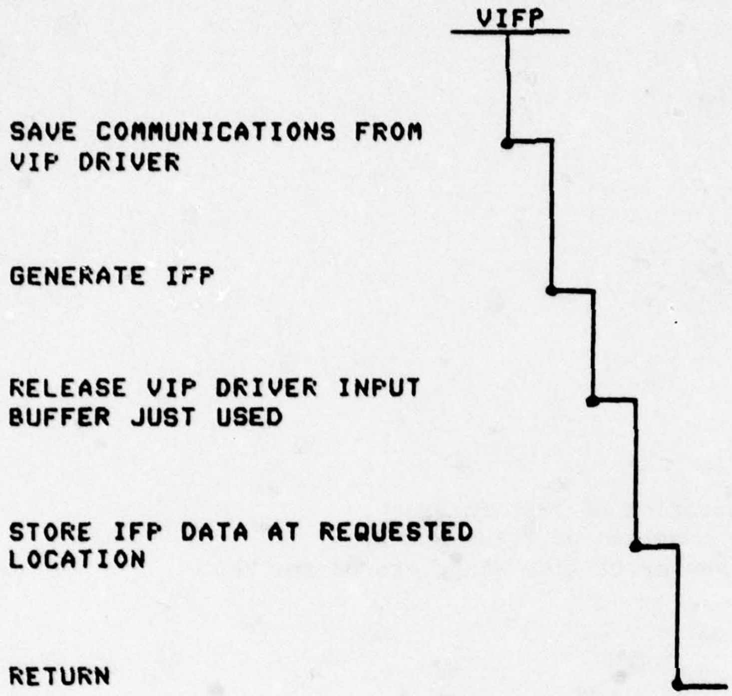
Cancels: None

Input arguments: IFP - Location of IFP storage
IVTIM - Location of YTIME storage
IVTS - Number of time slots stored for VRP

Common variables: None

Files created/changed: None

Files referenced: None



VIFP FLOWCHART (SHEET 1 OF 1)

PRESENT

Description: Presents phrase prompts on the requested prompt device.

Entry point: PRESENT

Classification: Subroutine

Period: None

Language: F

Activated/called by: COLLECT, VDCVAL1, VDCVAL2

Cancelled by: N/A

Activates/calls: CPU2 $TTO controller, IPBOUT2

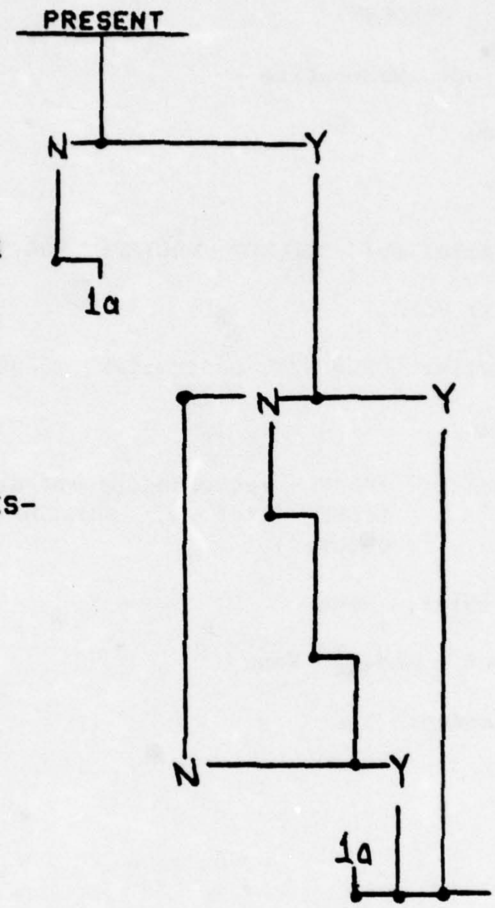
Cancels: None

Input arguments: IPDEV - requested prompt device: 1=$VRO, 2=CRT, 3=AUDIO
IPLST - list of phrases to be prompted (maximum 6 phrases)

Common variables: None

Files created/changed: None

Files referenced: None



IS PROMPT DEVICE TTO?

REQUEST PROMPT FROM CPU 1
DEVICE

IS PHRASE NUMBER ZERO?

OUTPUT PHRASE WHICH CORRES-
PONDS TO ARGUMENT

OUTPUT A PAUSE

IS THE ARGUMENT COUNT 6?

RETURN

PRESENT FLOWCHART (SHEET 1 OF 1)

FORMIT

Description: Forms voice reference patterns for the requested phrase.

Entry point: FORMIT

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: VGVRP

Cancels: None

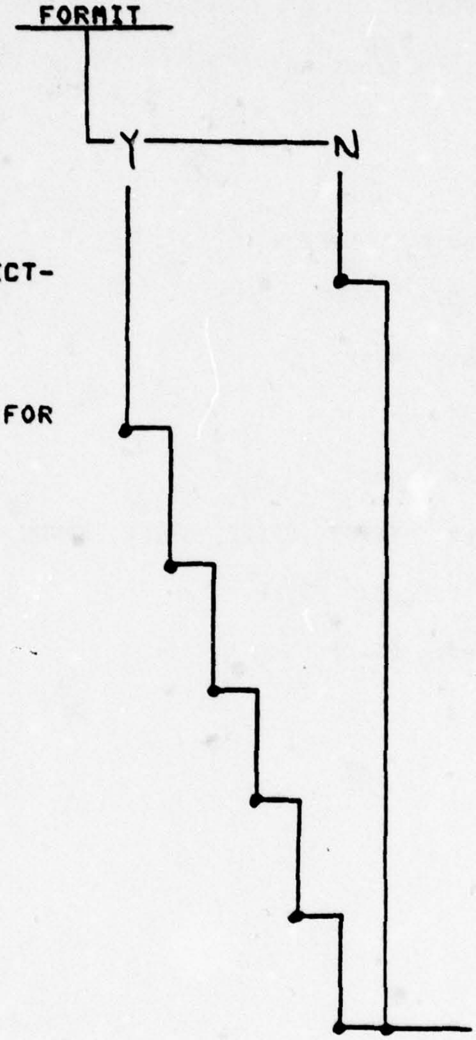
Input arguments: None

Common variables: SPLST, SPIFP, SPVRP, SPNUM, SPAT, VCFRAZ, SPID

Files created/changed: FNVRP

Files referenced: FNIFP

FORMIT



ARE THERE ENOUGH IFPS TO CREATE A VRP?

TYPE NUMBER OF IFPS COLLECTED

GET NUMBER OF TIME SLOTS FOR VRP

GET IFPS

FORM VRP

SET VRP COLLECTED FLAG

FILE VRP AWAY

RETURN

FORMIT FLOWCHART (SHEET 1 OF 1)

VGVRP

Description: Forms a voice reference pattern using the input feature patterns given.

Entry point: VGVRP

Classification: Subroutine

Period: None

Language: F

Activated/called by: FORMIT

Cancelled by: N/A

Activates/calls: None

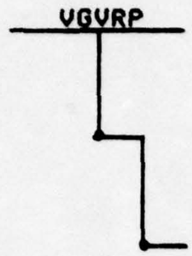
Cancels: None

Input arguments: IFPS - array if IFPs previously collected
NREP - number of repetitions
VRP-VRP array, IVTS-number of time slots for VRP

Common variables: VCWGT

Files created/changed: None

Files referenced: None



COMPUTE VRP

RETURN

VGVRP FLOWCHART (SHEET 1 OF 1)

VDCVAL1

Description: Validates the given phrases to the requested percentage accuracy. Prompts are re-issued until the percentage meets the requirements.

Entry point: VDCVAL1

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: BEGIN, TEST, TERMINATE

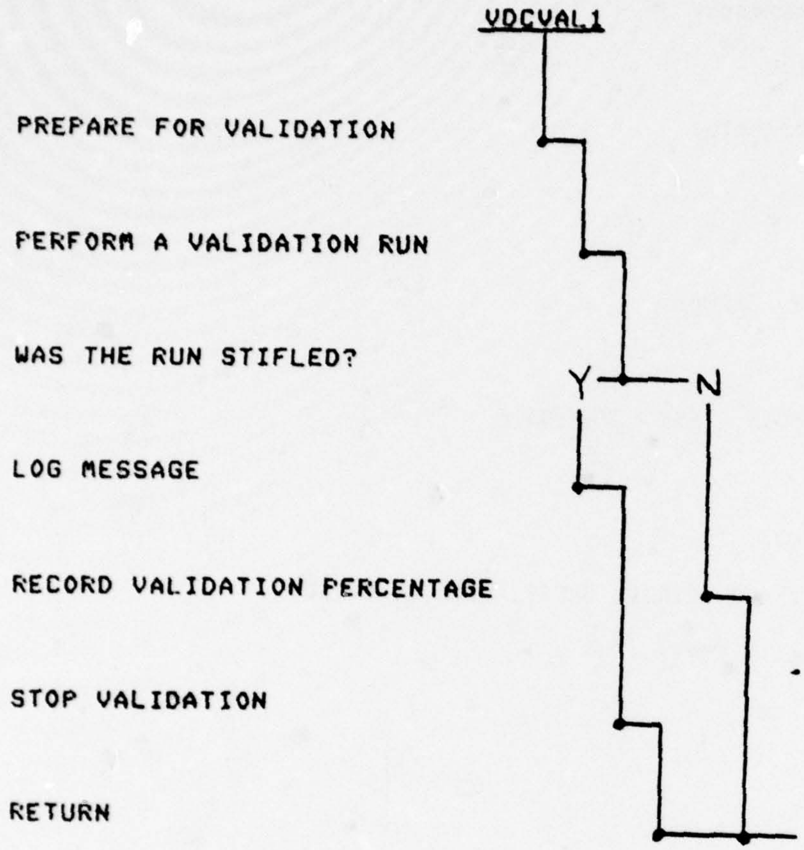
Cancels: None

Input arguments: None

Common variables: VLPCT, VLSTF, SPLST, VLARGS, SPVAL

Files created/changed: FNVRP

Files referenced: None



VDCVAL1 FLOWCHART (SHEET 1 OF 1)

TEST

Description: Performs a validation run for the given phrase(s). The run is terminated by the attainment of the requested validation percentage in three consecutive repeats or by student request.

Entry point: TEST

Classification: Subroutine

Period: None

Language: F

Activated/called by: VDCVAL1

Cancelled by: N/A

Activates/calls: VIPON, VIPOFF, PRESENT

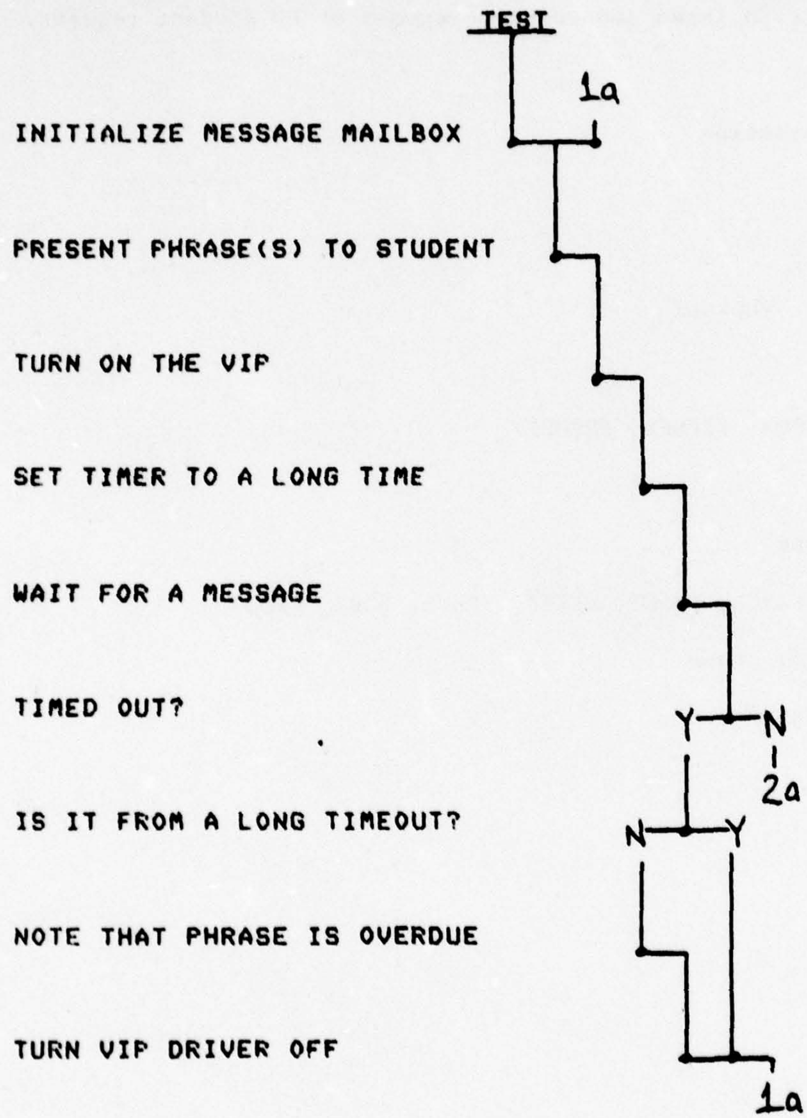
Cancels: None

Input arguments: None

Common variables: VLARGS, VLPCT, VLSTF, SPLVL, RCBF, BXCOG

Files created/changed: None

Files referenced: None



TEST FLOWCHART (SHEET 1 OF 3)

WAS THE MESSAGE MISUNDER-
STOOD?

WAS THE MESSAGE A STIFLE?

SET STIFLE FLAG

WAS THE MESSAGE TOO LONG OR
SHORT?

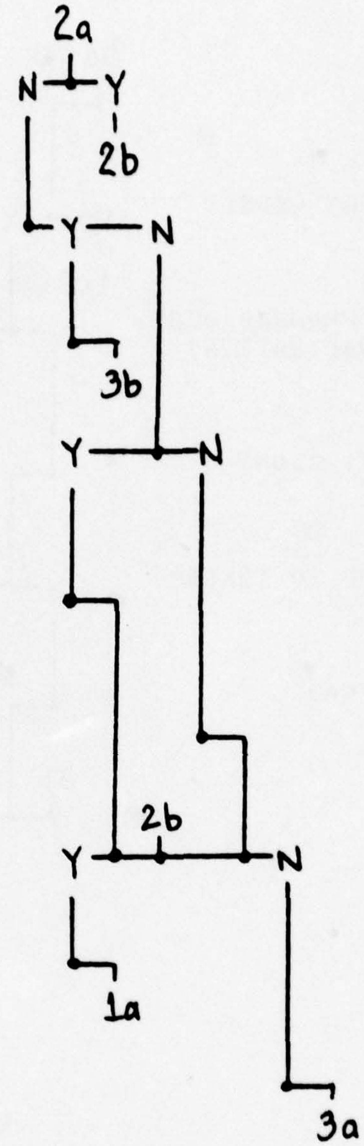
NOTE MESSAGE UNDERFLOW/OVER-
FLOW

STORE THE MESSAGE DATA

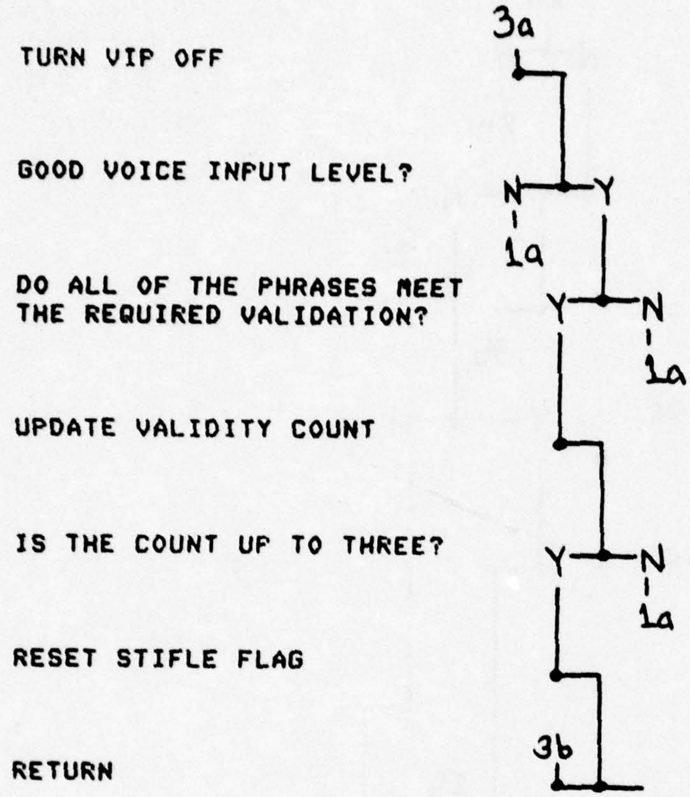
ARE MORE PHRASES EXPECTED?

SET A SHORT TIMEOUT

WAIT FOR STUDENT SILENCE



TEST FLOWCHART (SHEET 2 OF 3)



TEST FLOWCHART (SHEET 3 OF 3)

BEGIN

Description: Preparations for the voice validation mode are made by this routine.

Entry point: BEGIN

Classification: Subroutine

Period: None

Language: F

Activated/called by: VDCVAL1, VDCVAL2

Cancelled by: N/A

Activates/calls: VSRRC, user clock, STIFLE, VRPLD

Cancels: None

Input arguments: None

Common variables: VLSTF, RCRES, VLRES, EVVRPD

Files created/changed: None

Files referenced: None

BEGIN

LOAD ALL VRPS

SET RECOGNITION RESOLUTION
MASKS TO ACCEPT ALL PHRASES

INITIATE RECOGNITION TASK
WITH REQUIRED VALIDATION
PERCENTAGE

SET STIFLE FLAG FALSE

INITIATE STIFLE TASK

WAIT FOR VRPS TO BE LOADED

START USER CLOCK

RETURN

BEGIN FLOWCHART (SHEET 1 OF 1)

STIFLE

Description: Waits for STIFLE key input. The student selects STIFLE when he tires of the validation mode (or feels frustrated or antagonistic).

Entry point: STIFLE

Classification: Task

Period: None

Language: F

Activated/called by: BEGIN

Cancelled by: TERMINATE

Activates/calls: None

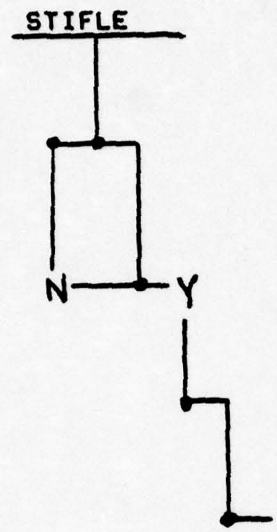
Cancels: Self

Input arguments: None

Common variables: EVKEY, BXCOG, KYLST

Files created/changed: None

Files referenced: None



WAIT FOR STUDENT KEYBOARD INPUT

IS THE INPUT A STIFLE?

SEND STIFLE MESSAGE

DIE

STIFLE FLOWCHART (SHEET 1 OF 1)

TERMINATE

Description: Cleans up for validation mode.

Entry point: TERMINATE

Classification: Subroutine

Period: None

Language: F

Activated/called by: VDCVAL1, VDCVAL2

Cancelled by: N/A

Activates/calls: None

Cancels: User clock, STIFLE, VSRRC

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

TERMINATE

KILL USER CLOCK

KILL STIFLE TASK

KILL VOICE RECOGNITION

TYPE END OF RUN MESSAGE

RETURN

TERMINATE FLOWCHART (SHEET 1 OF 1)

VRPLD

Description: Loads voice reference patterns into virtual memory.

Entry point: VRPLD

Classification: Task

Period: None

Language: F

Activated/called by: BEGIN, SUSON

Cancelled by: N/A

Activates/calls: None

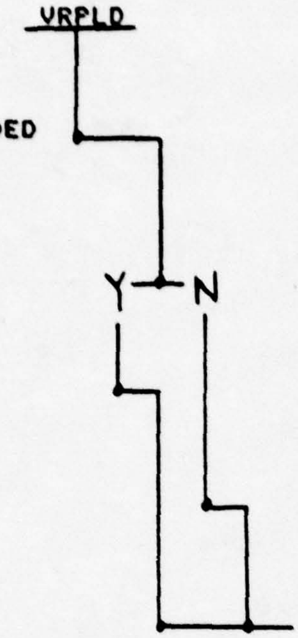
Cancels: Self

Input arguments: None

Common variables: SPLOD, EVVRPD

Files created/changed: None

Files referenced: FNVRP



READ THE VRPS INTO EXTENDED
MEMORY

ANY ERRORS?

TYPE ERROR MESSAGE

SEND DONE MESSAGE WAKEUP

DIE

VRPLD FLOWCHART (SHEET 1 OF 1)

VDCVAL2

Description: Validates student voice inputs for VRP defined phrases. Recognized phrases are mimicked after 1 second of silence.

Entry point: VDCVAL2

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: BEGIN, VIPON, FIPOFF, PRESENT, TERMINATE

Cancels: None

Input arguments: None

Common variables: SPDEV, SPLVL, RCBF, VLPCT, BXCOG

Files created/changed: None

Files referenced: None

VDCVAL2

SET VALIDATION PERCENTAGE TO ZERO

PREPARE FOR VALIDATION

TURN ON THE VIF

WAIT FOR A MESSAGE

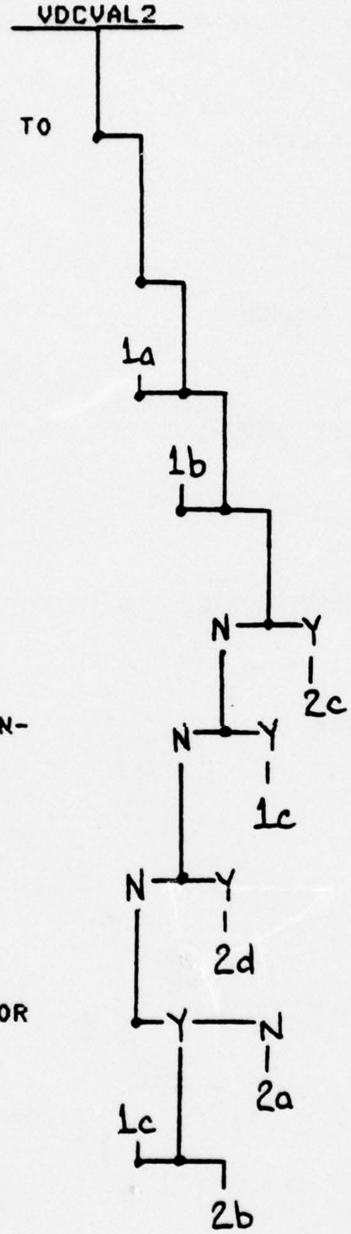
DID A TIMEOUT OCCUR?

WAS THE PHRASE INPUT MISUNDERSTOOD?

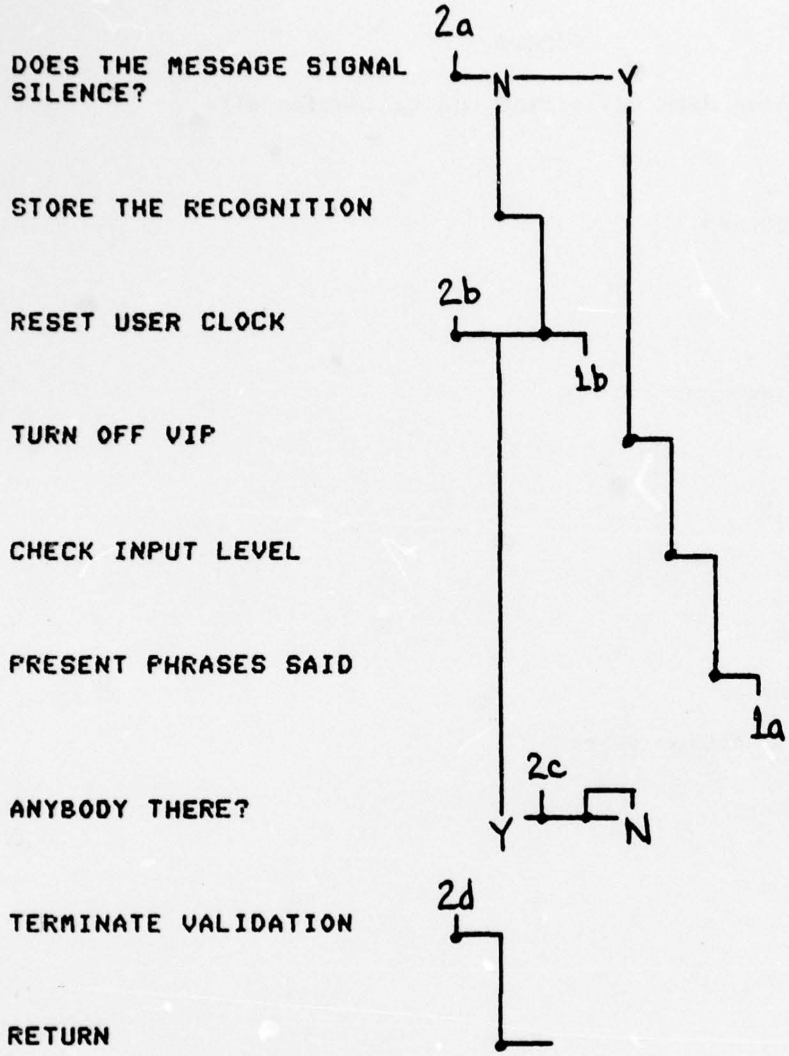
WAS THE MESSAGE A STIFLE?

WAS THE MESSAGE TOO LONG OR SHORT?

COMPLAIN TO THE SPEAKER



VDCVAL2 FLOWCHART (SHEET 1 OF 2)



VDCVAL2 FLOWCHART (SHEET 2 OF 2)

VDCOFF

Description: Turns voice data collection and validation off.

Entry point: VDCOFF

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: FNVRP, FNIFP

Files referenced: None

VDCOFF

PICK OUT PHRASES FOR WHICH IFPS BUT NOT VRPS HAVE BEEN DEFINED

LIST ALL SUCH PHRASES

OUTPUT UPDATED VRP AND IFP STATUS ARRAYS

CLOSE TRAINEE VOICE FILES

RETURN

VDCOFF FLOWCHART (SHEET 1 OF 1)

SUSON

Description: Activates and prepares all speech understanding files and tasks.

Entry point: SUSON

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: VSRRC, VIPON, VRPLD, SAID

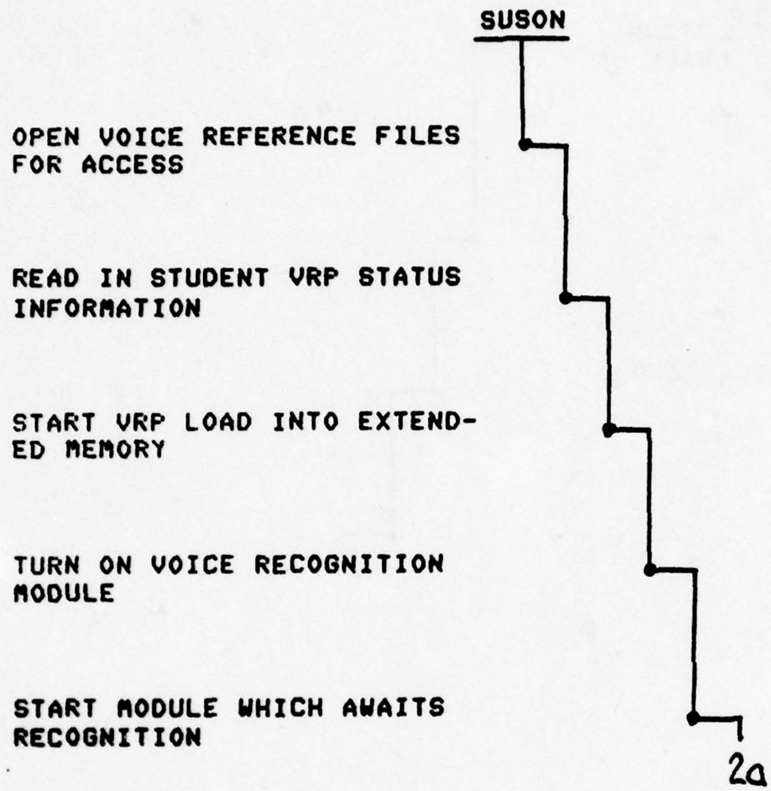
Cancels: None

Input arguments: None

Common variables: RCRES, RCPHS, EVVRPD

Files created/changed: None

Files referenced: FNVRP



SUSON FLOWCHART (SHEET 1 OF 2)

2a

SET RECOGNITION RESOLUTION MASKS TO SINGLE OUT PHASE OF FLIGHT INFORMATION

TURN VIP ON

WAIT FOR VRPS TO BE LOADED

RETURN

SUSON FLOWCHART (SHEET 2 OF 2)

SUSOFF

Description: Turns off all speech understanding modules. All speech files are closed.

Entry point: SUSOFF

Classification: Subroutine

Period: None

Language: F

Activated/called by: SPEECH

Cancelled by: N/A

Activates/calls: VIPOFF

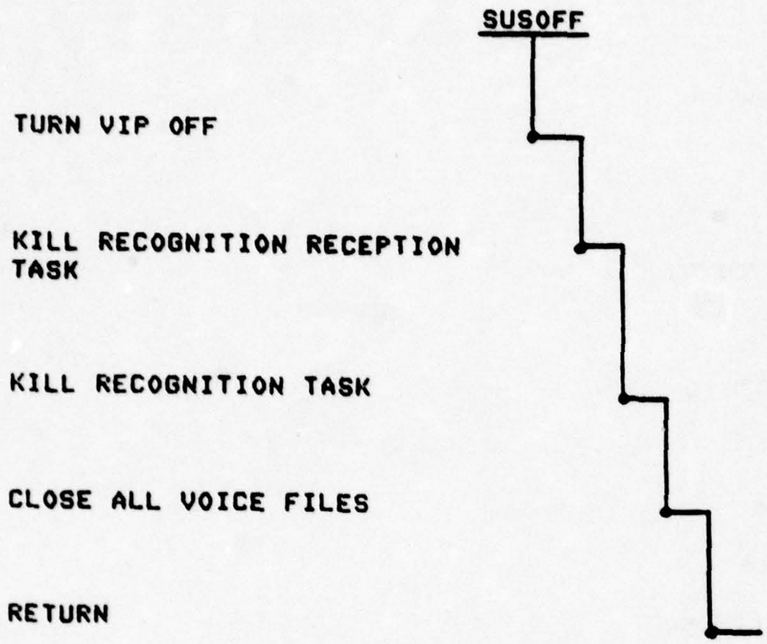
Cancels: VSRRC, SAID

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None



SUSOFF FLOWCHART (SHEET 1 OF 1)

Speech Recognition

Speech recognition (SR) compares an input phrase with stored VRPs to determine the identity of the phrases spoken. The identified phrases are then shipped to CPU1 for speech understanding processing or are identified to the validation module. In the first case, SR is activated by SUSON and in the latter by BEGIN.

Phrases involved in the recognition process and the recognition logic details follow.

Phrases for Recognition. All final controller phrases have been classified and categorized. The purpose of this classification is to aid the speech recognition module in its search for a likely voice reference pattern (VRP) as well as to provide a logical grouping for the phrases.

The classification of phrases produces an identification word which describes the phrase content and use. By applying a masking scheme on the identification word, particular types of phrases may be singled out. In general the recognition algorithm examines the phrases which are most likely to have been uttered (valid phrase types are provided by the final controller modules) and then proceeds to examine the remainder of the phrases if a good recognition is not produced.

All interprogram communication with the voice recognition, speech understanding, and voice data collection modules reference phrases either by the sequence order or by the phrase identification. The sequence number is in decimal notation and reflects the order of the provided list. The identification word is expressed in octal notation.

Identification Words. The identification words are the result of bit settings which correspond to the phrase content and use. Each phrase is assigned 1 16-bit word with bits delegated as shown in table 13.

Bit Setting Definitions. The bits of the identification word are either set to 1 or are reset to 0 to indicate the following traits:

Number of time slots for VRPs is a storage reference since all phrases with three syllables or less shall be stored in 16 time slots instead of the traditional 32. Bit 0 is set to indicate 32 time slots.

Key phrases are those phrases which can stand alone, e.g., digits are not key phrases since they must be used in conjunction with headings or a wind message. Bit 1 is set for a key phrase.

Phase 1 refers to the beginning of the GCA in which the handoff to the final controller is made. The final controller speaks to either the pilot or the pattern controller during parts of this phase. The bit is set if the phrase is valid during this phase.

TABLE 13. SPEECH RECOGNITION IDENTIFICATION WORD BITS

BIT		DEFINITION
0	Number of time slots for VRPs:	0 = 16 1 = 32
1	Key Phrase	0 = NO 1 = YES
2	Phrase Used In:	1 = Phase 1
3		1 = Phase 2
4		1 = Phase 3
5	Phrase Type:	1 = Glidepath
6	IF 0 = N/A	1 = Course
7		1 = Heading
8		1 = Range
9		1 = Missed approach, waveoff, impending missed approach, wind clearance
10		1 = No Gyro
11		1 = Trend for GP/Course/Land
12	Individual Phrase ID within category OR GP/Course/Heading Characteristics	GP/Course/Heading characteristics: 1 = Above, Right, TLH; 0 = Below or on, Left of Course, TRH
13		Plane is: 1 = Slightly
14		1 = Between slightly and well
15		0 = on, 1 = well

TLH = turn left heading
TRH = turn right heading
GP = glidepath

Phase 2 references the part of the approach which follows the initial hand-off. It is defined to begin after the "begin descent" advisory. This bit is set if the phrase is valid during phase 2.

Phase 3 begins as decision height, radar contact lost, or a missed approach execution or waveoff is announced. Such phrases which accomplish this transition are also classified as legal in the phase which they terminate. For example, "at decision height" is classified as a phase 2 and 3 phrase. Phase 3 continues until the final controller has completed the final handoff.

Glidepath is set for all glidepath related messages. This also, applies for "too low for safe approach," etc.

Course is set for all course related messages.

Heading is set for all heading messages or parts of heading messages, namely the digits.

Range is set for all range related messages, i.e., "X miles from touchdown."

Missed approach/wind is set for all phrases associated with a missed approach or a wind or clearance message. This includes all types of wave-offs, also "radar contact lost" and others which imply a pending waveoff are included.

No-gyro is set for no-gyro advisories; no-gyro type turns inclusive.

Trend/land is set for either glidepath or course trend messages or advisories which are associated with an aircraft's landing, e.g., "contact tower after landing."

Bits 12-15:

Individual phrase id values are assigned to differentiate between phrases with the same traits expressed in bits 1 through 11. This is not the case, however, for glidepath, course, and their associated trend messages and heading messages.

For glidepath, course, and heading bits are assigned in the following manner:

Bit 12 is set for:

Glidepath messages which apply when the plane's center is above the glidepath

Course messages which apply when the plane's center is to right of course, including turn left since turn left is associated with being to the right of course

Trend messages used while plane satisfies the above glidepath or right of course course condition.

Bit 13 is set for:
Being slightly above/below glidepath, slightly right/left of course, and for corresponding trends.

Bit 14 is set for:
Being between slightly and well, e.g., "below glidepath," "right of course," and its use for trends is valid for the same conditions.

Bit 15 is set for:
Being well above, below, right, or left and the same for trend validity.

Example for bits 12-15:
"Going further above glidepath" is used when:

Bit 12 = 1, the plane is above the glidepath
13 = 0, it is past the slightly zone
14 = 1, it is transitting to well above zone
15 = 1, the plane is going through the well zone

Phrase List. The GCA-CTS phrases are shown in table 14. The order in which the phrases are listed reflects the logical groups. This was produced by a sort on the last 5 digits of the phrase identification word (since the first digit is merely a VRP length indicator).

Additional identification words are provided for unrecognizable phrases or low input level.

000001	Message not understood
000002	Message too short
000003	Message too long
000004	Input level low

Speech Recognition Logic. The speech recognition logic proceeds in a fairly sequential manner. When the speech recognition routine VSRRC is activated, it awaits notification of a voice input from the Threshold 500 driver, VIPDR. The VIPDR fills an available input buffer and sends a communication packet to VSRRC. Figure 18 displays the SR structure.

Input Buffers. SR maintains two distinct input buffers, A and B. Since a sample consists of 32 bits of information and Eclipse words hold 16 bits of information, each of these identical buffers consists of two parts: part 1 holds the first 16 bits of a sample and part 2 holds the last 16 bits, as shown in figure 19.

The size of each buffer, BFSZ, will be a function of the length of the longest phrase to be recognized. Since a set of 32 features (a sample) is provided every 2 milliseconds, the length of each part of the buffer should be N/2 words long, where N is the number of milliseconds in the longest phrase. This value is an assembly parameter.

TABLE 14. ORDERED LIST OF GCA-CTS PHRASES

Reference Number	ID	Phrase
1	004302	1 MILE.
2	104303	1 AND 1/2 MILES.
3	004304	2 MILES.
4	104305	2 AND 1/2 MILES.
5	004306	3 MILES.
6	104307	3 AND 1/2 MILES.
7	010101	AT.
8	010114	TWELVE.
9	010117	FIFTEEN.
10	010124	TWENTY.
11	010131	TWENTY-FIVE.
12	010136	THIRTY.
13	034500	0
14	034501	1
15	034502	2
16	034503	3
17	034504	4
18	034505	5
19	034506	6
20	034507	7
21	034510	8
22	034511	9
23	144021	CONTACT TOWER AFTER LANDING.
24	144024	BUTTON 4, CLEAR.
25	144026	BUTTON 6, CLEAR.
26	044100	MISSED APPROACH
27	144101	IF RUNWAY NOT IN SIGHT.
28	144102	IF RUNWAY NOT IN SIGHT, EXECUTE MISSED APPROACH.
29	144103	IF RUNWAY NOT IN SIGHT, CLIMB AND MAINTAIN 1500.
30	044104	BUTTON 4.
31	144105	PROCEED DIRECT POINT BRAVO, HOLD UNTIL ADVISED BY GCA.
32	044106	BUTTON 6.
33	044110	ON THE GO.
34	144220	OVER LANDING THRESHOLD.
35	145000	TOO FAR LEFT FOR SAFE APPROACH.
36	145010	TOO FAR RIGHT FOR SAFE APPROACH.
37	146001	TOO LOW FOR SAFE APPROACH.
38	146011	TOO HIGH FOR SAFE APPROACH.
39	050100	WIND.
40	150102	CLEARED FOR LOW APPROACH.
41	150103	CLEARED FOR TOUCH AND GO.
42	050104	CLEARED TO LAND.
43	150201	1 MILE FROM TOUCHDOWN.
44	150202	2 MILES FROM TOUCHDOWN.
45	150203	3 MILES FROM TOUCHDOWN.
46	150204	4 MILES FROM TOUCHDOWN.
47	151001	WELL LEFT OF COURSE.
48	051002	LEFT OF COURSE.
49	151011	WELL RIGHT OF COURSE.
50	051012	RIGHT OF COURSE.

TABLE 14. ORDERED LIST OF GCA-CTS PHRASES (CONT)

51	152001	WELL BELOW GLIDEPATH.
52	152011	WELL ABOVE GLIDEPATH.
53	152023	GOING FURTHER BELOW GLIDEPATH.
54	152033	GOING FURTHER ABOVE GLIDEPATH.
55	154103	CLIMB AND MAINTAIN 1500.
56	154200	AT DECISION HEIGHT.
57	055000	ON COURSE.
58	155004	SLIGHTLY LEFT OF COURSE.
59	155014	SLIGHTLY RIGHT OF COURSE.
60	055033	CORRECTING.
61	056000	ON GLIDEPATH.
62	156002	BELOW GLIDEPATH.
63	156004	SLIGHTLY BELOW GLIDEPATH.
64	156012	ABOVE GLIDEPATH.
65	156014	SLIGHTLY ABOVE GLIDEPATH.
66	156024	GOING BELOW GLIDEPATH.
67	056027	COMING UP.
68	156034	GOING ABOVE GLIDEPATH.
69	056037	COMING DOWN.
70	160001	POSITION 1 ROGER
71	160004	RADAR BUTTON 4
72	160006	RADAR BUTTON 6
73	160007	THIS IS YOUR FINAL CONTROLLER, HOW DO YOU HEAR ME?
74	160010	WHEELS SHOULD BE DOWN.
75	160011	DO NOT ACKNOWLEDGE FURTHER TRANSMISSIONS.
76	160012	APPROACHING GLIDEPATH.
77	160013	BEGIN DESCENT.
78	160014	GIVE ME BUTTON 4.
79	160016	GIVE ME BUTTON 6.
80	164001	ARMY 876
81	164002	MARINE 687
82	164003	NAVY 310
83	164004	AIR FORCE 307
84	064010	OVER
85	170040	THIS WILL BE A NO GYRO APPROACH.
86	170041	MAKE HALF STANDARD RATE TURNS.
87	170205	5 MILES FROM TOUCHDOWN.
88	170206	6 MILES FROM TOUCHDOWN.
89	170207	7 MILES FROM TOUCHDOWN.
90	170210	8 MILES FROM TOUCHDOWN.
91	172001	LOW ALTITUDE ALERT, CHECK YOUR ALTITUDE IMMEDIATELY.
92	174000	HOW DO YOU HEAR ME NOW?
93	074000	CORRECTION
94	074043	TURN RIGHT.
95	074047	STOP TURN.
96	074053	TURN LEFT.
97	174100	EXECUTE MISSED APPROACH.
98	174101	RADAR CONTACT LOST.
99	174102	CLIMB AND MAINTAIN 3000
100	174403	TURN RIGHT HEADING.
101	074407	HEADING.
102	174413	TURN LEFT HEADING.

NOTE: It is assumed that the aircraft will not stray into the well zones or any further than slightly off course during phase 3 because the pilot will take over visually at decision height.

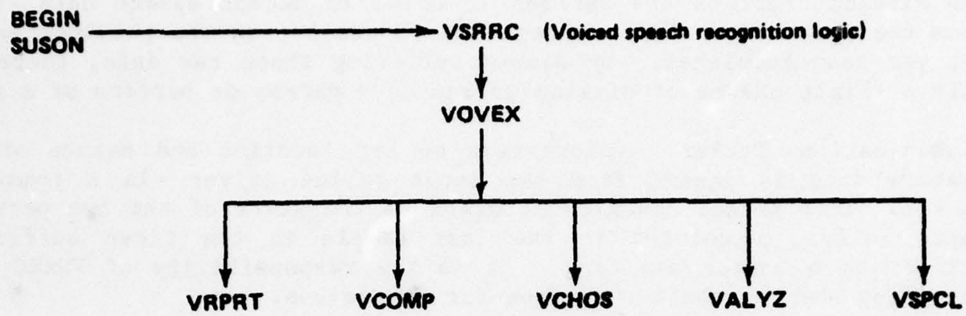


Figure 18. Voiced Speech Recognition Routines

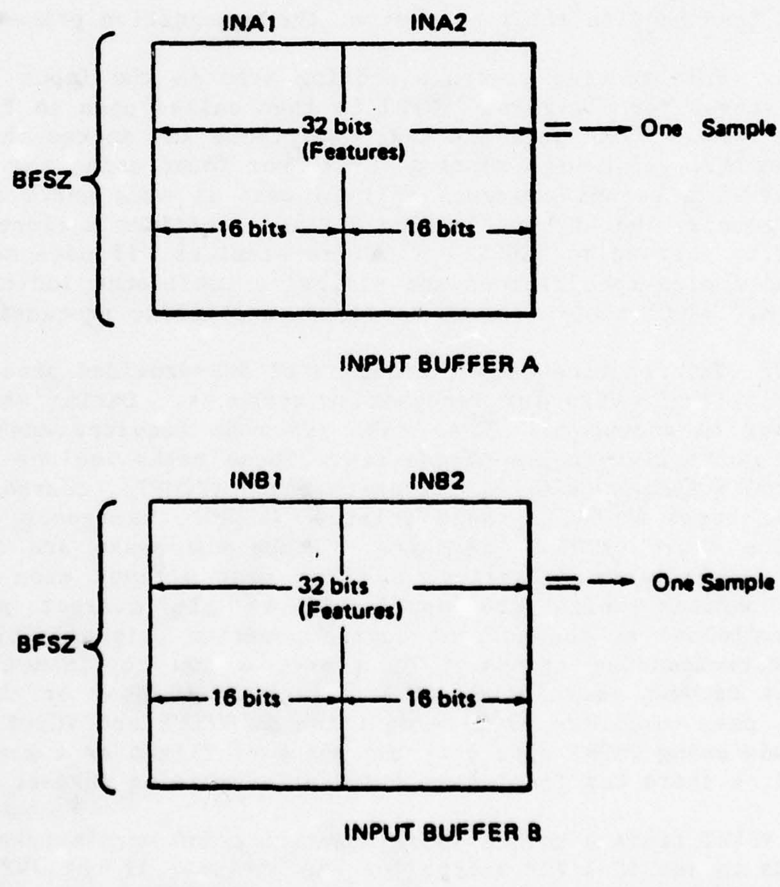


Figure 19. Speech Recognition Input Buffer Structure

Two distinct buffers are defined in order to accept speech data (features) from the preprocessor even though the processing on the previous phrase has not yet been completed. By double buffering these raw data, there will be only a slight chance of missing or losing a phrase or portion of a phrase.

Communications Packet. Information on the location and nature of the raw feature data is passed from the input device driver via a communication packet. This packet contains pointers to the start of the two parts of the input buffer, a pointer to the last sample in the first buffer, and a pointer to a buffer use flag. It is the responsibility of VSRRC to clear this flag when the buffer is free for future use.

Another word in the communications packet is reserved for unrecognizable phrase error returns for instances such as buffer overflows or phrases which are too short. The LP_4 time in half second clock ticks and 100 millisecond increment offsets is also sent via the communications packet.

VSRRC. This routine awaits notification of a voice input and forms IFPs if an error return is not received. A 32 time slot IFP is formed as well as a 16 time slot IFP. These are stored in the input buffer area. Subroutine VOVEX then handles the remainder of the recognition process.

VOVEX. This routine clears a scoring area in the input buffer to accommodate scores for all VRPs. VRPRT is then called upon to find a set of plausible VRPs. VCOMP performs the comparisons and scores the VRPs in relation to the IFP. If a high scoring VRP is not found using the first set provided by VRPRT, a second and even a third pass is made until a matching VRP has been found. The highest scoring VRP and sometimes a close second choice are found by subroutine VCHOS. VALYZ re-examines, if necessary, the first and second choice recognitions and assigns a confidence indicator to the recognition. VSPCL then performs application specific processing.

VRPRT. This routine uses validation of SUS-provided phrase masks to locate sets of likely VRPs for recognition purposes. During validation the masks are set to accept all VRPs. The SUS mode requires masks provided by the model controller-chosen phrase set. These masks include glidepath position related messages (RCGPP), glidepath trend (RCGPT), course position (RCCRP), course trend (RCCRT), range related (RCRNG), emergency related (RCEMERG), and the other (RCOTHR) messages. Since the masks are set to appropriate final controller advisories, a first pass attempt made by VRPRT includes only phrases which are applicable to the correct glidepath position (above/below) or the correct course position (right/left) and likewise for the corresponding trends. Therefore, a low confidence conflict may not result between above/below or left/right. If none of the phrases in this first pass qualifies as a recognition in VCOMP and VCHOS, a second attempt is made using VRPRT with only the phase of flight as a mask. Again, if this fails, a third and final pass finds all remaining phrases yet untried.

When VRPRT finds a phrase whose identification word matches a mask, it first checks to see if a VRP exists for the phrase. If not VRP exists, the score is set to one. The two's complement of the phrase VRP record number is set

in the score word of the phrase if a VRP does exist and a score has not already been computed. One complete pass through the identification words is made each time VRPRT is called.

VCOMP. This routine scans the score table for negative entries. The record number is translated to a block and word offset. This information is used to map in the necessary block which holds the VRP into the window buffer before the IFP/VRP comparison is made.

Since the VRPs are loaded into virtual memory when SUS is turned on (SUSON), a window map scheme is necessary to access the VRPs. For this purpose a 1024-word buffer, VRPBF, has been defined as the window.

In VCOMP, the IFP is compared to each of the VRPs. This is done on a bit-by-bit basis, by the TTI provided high speed correlator (HSC), using the algorithm given in the table 15. The number (score) for each bit-by-bit comparison of the IFP and VRP is totaled and saved for each VRP. This calculation is repeated for each IFP/VRP pair, but with IFP first shifted up by one time slot and then again down by one time slot; the largest of the three scores is saved. All of these scores are normalized by the highest possible score, that is, the score that is obtainable if the IFP matched all VRP bits. If a negative score is obtained, it is set to 1 in the score table.

TABLE 15. IFP/VRP COMPARISON ALGORITHM

	Feature Set in VRP	Feature Not Set in VRP
Feature Set in IFP	2	-1
Feature Not Set in IFP	-1	0

VCHOS. The selection logic in VCHOS includes the routine which chooses the highest scores from the score table to build a choice table, CHOT. Another routine reorders CHOT so that the highest index/score pair is the first entry. A more basic procedure uses the aforementioned routines to extract the highest score together with a second choice score if one exists. A flag is set if a close second choice is found.

VALYZ. The analysis provided in VALYZ consists mainly of the Breaux test or second look. Following a scheme devised by Dr. Breaux of NAVTRAEQUIPCEN, the two VRPs which are in contention for first choice recognition are compared to find those time slots which are significantly different. These rows are then correlated with the corresponding rows of the IFP. The technique effectively causes the pattern recognition algorithm to weigh the distinctive portions of the utterance more heavily than the similar portions. CHOT is reordered after this test and the confidence level is again investigated.

VSPCL. Special recognition processing is provided for phrase groups such as headings, wind, or missed approach phrases. Special masks are sent to RCGPP, RCGPT, and RCCRP for easy recognition of anticipated phrases. In the case of headings and wind these masks single out the digits (and later "at" and wind speed for wind) for a first recognition pass. This boosts the probability of accumulating a set of phrases that concatenate to form a sensible message.

VSRRC

Description: Voiced speech recognition task awaits student inputs to be processed for recognition.

Entry point: VSRRC

Classification: Task

Period: None

Language: A

Activated/called by: BEGIN, SUSON

Cancelled by: N/A

Activates/calls: VOVEX

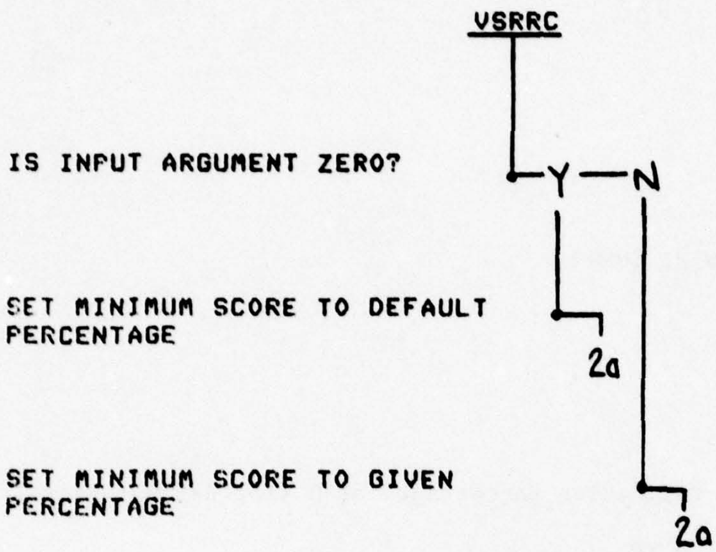
Cancels: None

Input arguments: ARG1 - Validation percentage, or 0 (for default score)

Common variables: SPFLG, RCBF

Files created/changed: None

Files referenced: None



VSRRC FLOWCHART (SHEET 1 OF 2)

WAIT FOR MESSAGE FROM VIP

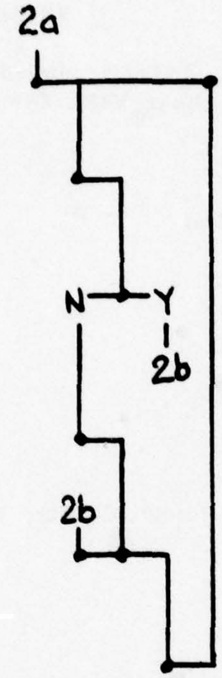
SAVE MESSAGE PACKET

IS MESSAGE TOO LONG OR TOO SHORT?

GENERATE IFP

DO RECOGNITION LOGIC

RELEASE INPUT BUFFER FOR REUSE



VSRRC FLOWCHART (SHEET 2 OF 2)

VOVEX

Description: VOVEX finds likely candidates for recognition, then compares and scores these VRPs for the given IFP.

Entry point: VOVEX

Classification: Subroutine

Period: None

Language: A

Activated/called by: VSRRC

Cancelled by: N/A

Activates/calls: VRPRT, VCOMP, VCHOS, VALYZ, VSPCL

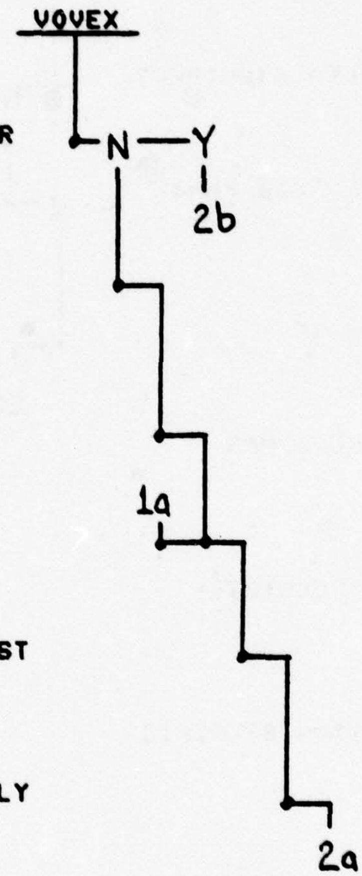
Cancels: None

Input arguments: None

Common variables: RCRES, RCFZIS, RCGPP

Files created/changed: None

Files referenced: None



IS THE MESSAGE TOO LONG OR SHORT?

FIND END OF IFP DATA IN BUFFER

CLEAR A SCORING AREA

FIND LIKELY VRPS

COMPARE IFPS WITH VRPS JUST FOUND

CHOOSE AND RANK MOST LIKELY PHRASE(S)

VOVEX FLOWCHART (SHEET 1 OF 2)

IS THE FIRST PLACE PHRASE
RANKED HIGHLY ENOUGH?

HAVE ALL VRPS BEEN EXAMINED?

HAVE ALL VRPS OF THIS PHASE
BEEN EXAMINED?

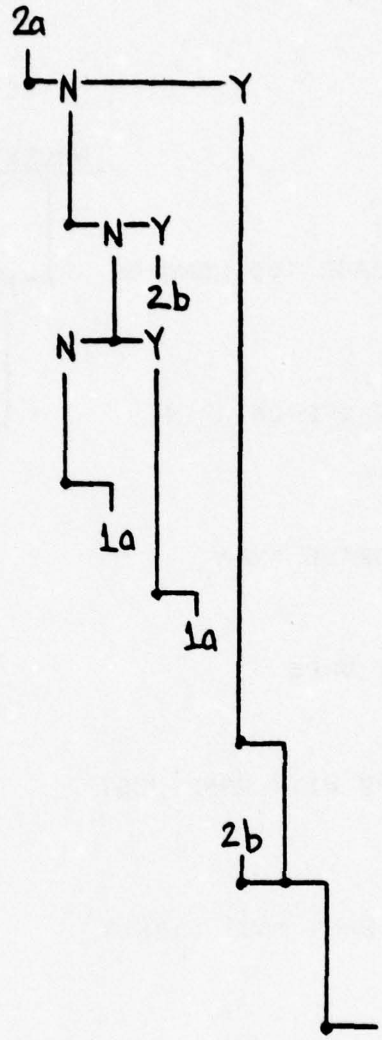
SET MASKS FOR PHASE

SET MASKS FOR ALL OTHER
PHRASES

ANALYZE TOP TWO POSSIBIL-
ITIES

PERFORM APPLICATION SPECIFIC
PROCESSING

RETURN



VOVEX FLOWCHART (SHEET 2 OF 2)

VRPRT

Description: Any VRP whose identification tag matches a model controller selection for the bits set in the associated resolution mask, is flagged with its location pointer. VRPRT returns a count of phrases flagged.

Entry point: VRPRT

Classification: Subroutine

Period: None

Language: A

Activated/called by: VOVEX

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: RCGPP, RCGPT, RCCRP, RCCRT, RCRNG, RCOTHR, SPVRP, SPID, RCRES, RCEMERG

Files created/changed: None

Files referenced: None

INITIALIZE

IS THE MESSAGE MASK GOOD?

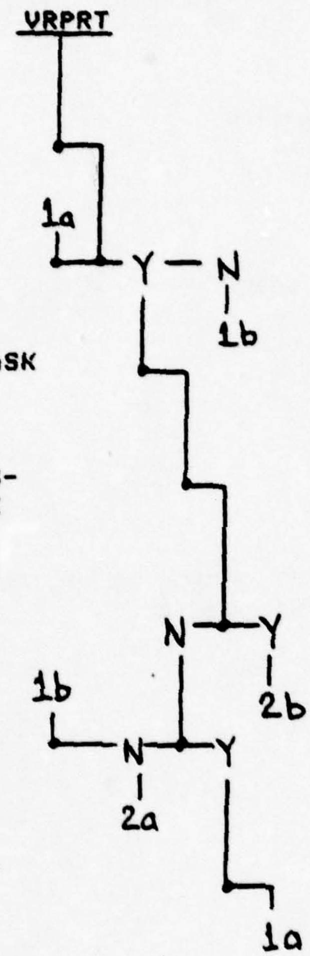
MASK ID WITH RESOLUTION MASK

COMPARE RESULT WITH CORRES-
PONDING MESSAGE TYPED MASK

DOES IT MATCH?

IS THERE ANOTHER MASK SET
THAT SHOULD BE APPLIED?

UPDATE POINTER TO NEXT SET
OF MASKS



VRPRT FLOWCHART (SHEET 1 OF 2)

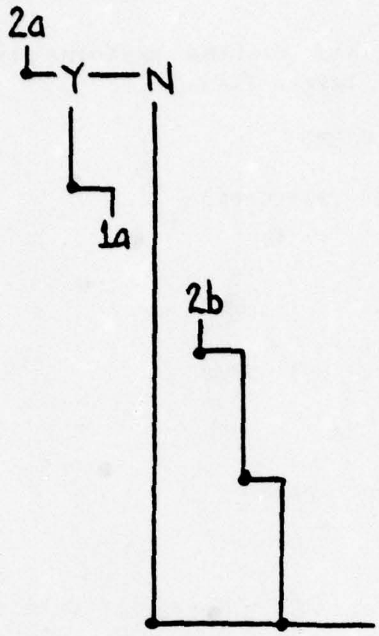
IS THERE ANOTHER PHRASE?

UPDATE POINTER TO NEXT
PHRASE

CALCULATE VRP INDEX

PLACE INDEX IN APPROPRIATE
SLOT IN SCORE AREA

RETURN



VRPRT FLOWCHART (SHEET 2 OF 2)

VCOMP

Description: This routine performs the comparison between the IFP and flagged VRPs.

Entry point: VCOMP

Classification: Subroutine

Period: None

Language: A

Activated/called by: VSRRC

Cancelled by: N/A

Activates/calls: None

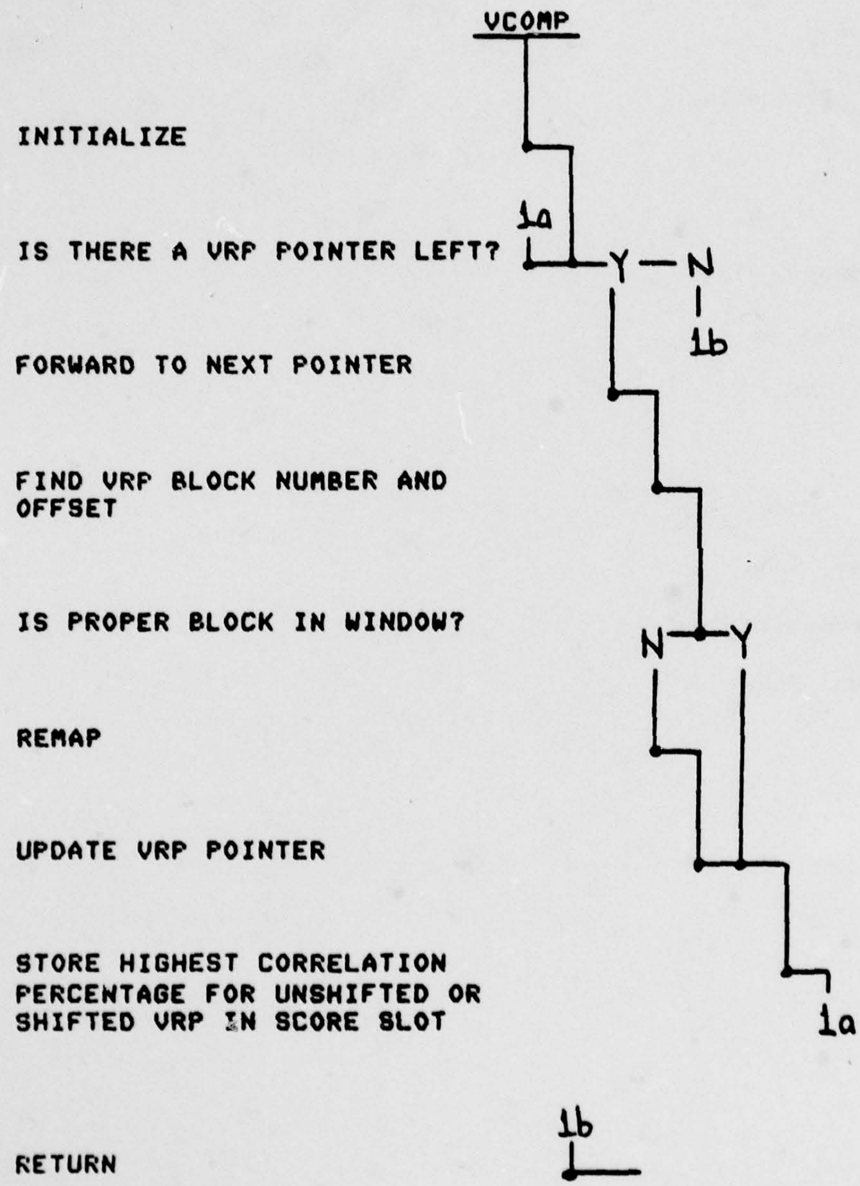
Cancels: None

Input arguments: AC1 - Number of VRPs to be investigated

Common variables: None

Files created/changed: None

Files referenced: None



VCOMP FLOWCHART (SHEET 1 OF 1)

VCHOS

Description: The highest scores are chosen from among the scores computed. A minimum score must be met. A second choice is reported if it exceeds the minimum score and is close to the winner.

Entry point: VCHOS

Classification: Subroutine

Period: None

Language: A

Activated/called by: VSRRC

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

VCHOS

INITIALIZE

FIND HIGHEST SCORE

IS IT ABOVE THE MINIMUM SCORE?

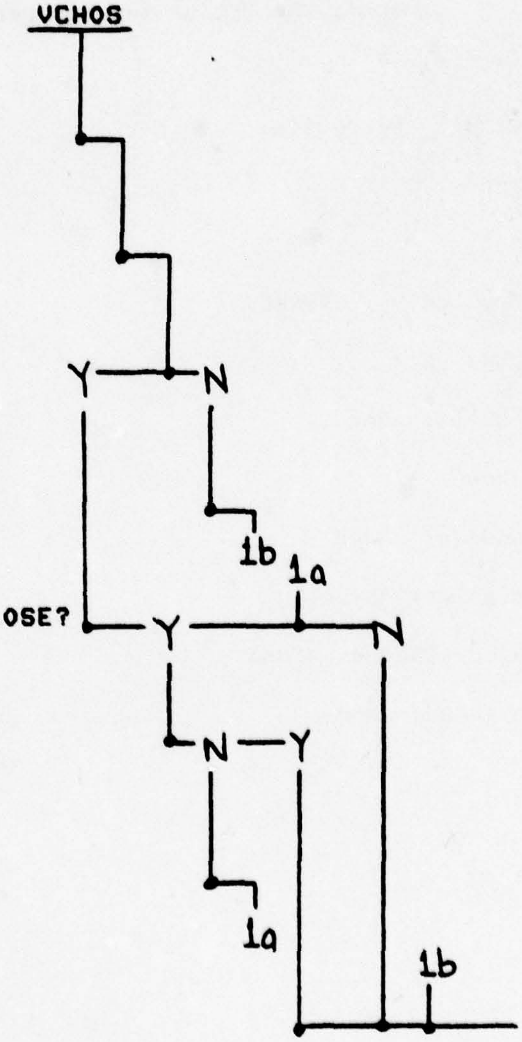
ERROR EXIT

DID ANOTHER SCORE COME CLOSE?

HAVE ENOUGH CHOICES BEEN FOUND?

FIND NEXT HIGHEST SCORE ABOVE MINIMUM SCORE

RETURN



VCHOS FLOWCHART (SHEET 1 OF 1)

VALYZ

Description: This routine analyzes the recognition choice(s). If only one choice was made, no further analysis is done and a high confidence factor is assigned to the recognition. If there are two choices, the Breaux test is performed.

Entry point: VALYZ

Classification: Subroutine

Period: None

Language: A

Activated/called by: VSRRC

Cancelled by: N/A

Activates/calls: None

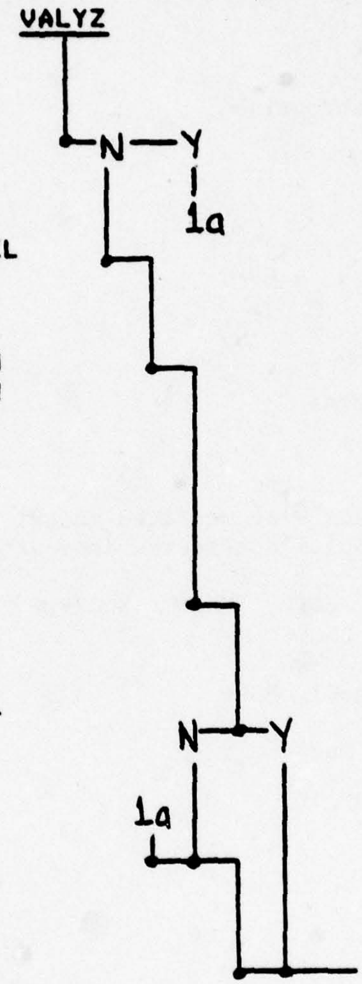
Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None



WAS ONLY ONE CHOICE MADE?

ASSUME LOW CONFIDENCE LEVEL

PERFORM BREAUX TEST, WHICH WEIGHS PARTS OF THE CHOSEN VRPS THAT DIFFER, MORE HEAVILY

REORDER CHOICE TABLE

IS THE SECOND CHOICE STILL CLOSE?

SET HIGH CONFIDENCE LEVEL

RETURN

VALYZ FLOWCHART (SHEET 1 OF 1)

VSPCL

Description: This routine performs vocabulary specific processing. Special message types which incorporate variable numerical phrases are identified and masks are prepared for their reception.

Entry point: VSPCL

Classification: Subroutine

Period: None

Language: A

Activated/called by: VSRRC

Cancelled by: N/A

Activates/calls: None

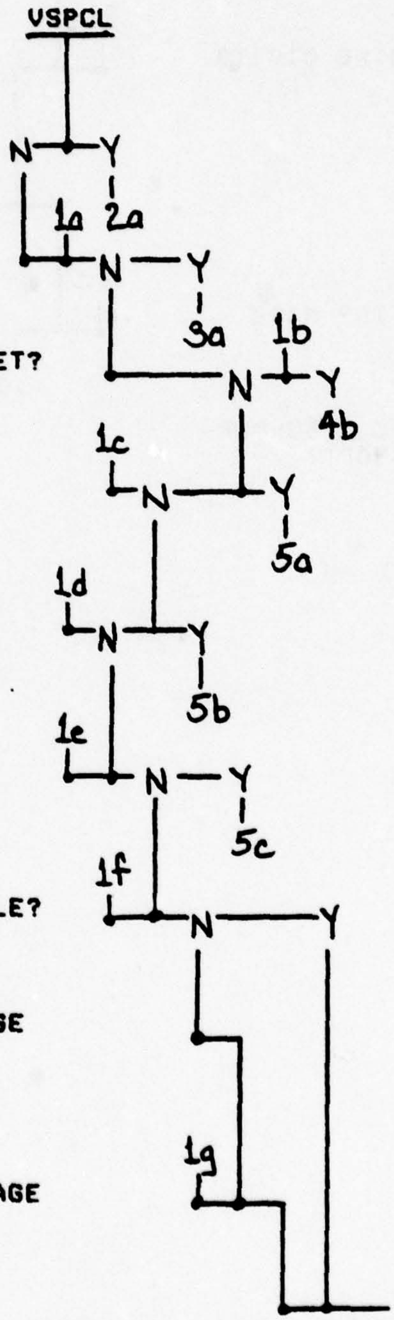
Cancels: None

Input arguments: AC0 - chosen item index
AC1 - confidence indication

Common variables: RCGPP, RCGPT, RCCRP, RCRES, RCBF(5), RCBF(6), RCBF(8), BXCOG

Files created/changed: None

Files referenced: None



IS HEADING FLAG SET?

IS WIND FLAG SET?

IS MISSED APPROACH FLAG SET?

IS RECOGNIZED PHRASE OR SECOND A HEADING PHRASE?

IS RECOGNIZES PHRASE OR SECOND "WIND"?

IS RECOGNIZED PHRASE OR SECOND "MISSED APPROACH"?

IS THE MESSAGE RECOGNIZABLE?

SET RECOGNITION TO "MESSAGE NOT UNDERSTOOD" OR "TOO LONG" OR "TOO SHORT."

SEND OUT RECOGNITION MESSAGE

RETURN

VSPCL FLOWCHART (SHEET 1 OF 5)

IS THIS A DIGIT?

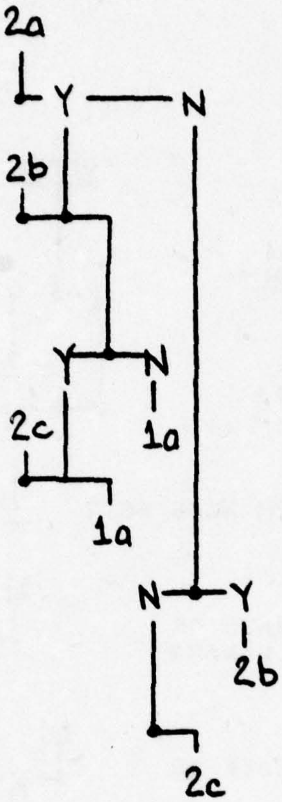
INCREMENT HEADING DIGITS
COUNT FLAG

IS COUNT 3?

RESET RECOGNITION MASK

WAS THE MESSAGE MISUNDER-
STOOD OR TOO SHORT?

RESET HEADING FLAG



VSPCL FLOWCHART (SHEET 2 OF 5)

IS THIS A DIGIT?

INCREMENT WIND DIGITS COUNT FLAG

IS COUNT 3?

SET RECOGNITION MASK TO ACCEPT "AT."

IS COUNT 4?

SET COUNT AND RECOGNITION MASK FOR WIND SPEED

INCREMENT COUNT

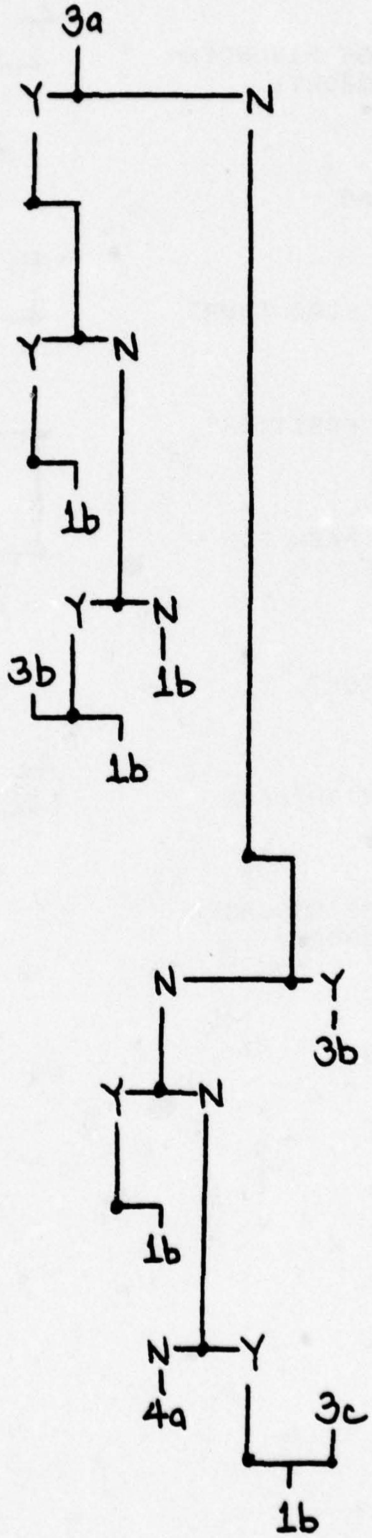
IS THIS "AT"?

IS THIS A WIND SPEED?

SET COUNT AND RECOGNITION MASK FOR CLEARANCE

IS THIS A CLEARANCE?

RESET RECOGNITION MASK



VSPCL FLOWCHART (SHEET 3 OF 5)

WAS THE MESSAGE MISUNDERSTOOD OR TOO SHORT?

RESET WIND FLAG

INCREMENT MAP FLAG COUNT

IS THIS A MAP POSITION?

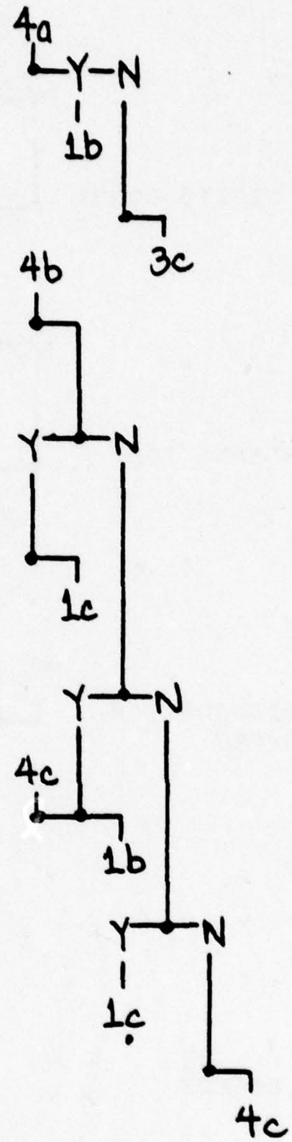
SET COUNT AND MASK FOR BUTTON MESSAGE

IS THIS A BUTTON?

RESET RECOGNITION MASK

WAS THE MESSAGE MISUNDERSTOOD OR TOO SHORT?

RESET MAP FLAG



VSPCL FLOWCHART (SHEET 4 OF 5)

SET HEADING FLAG TO ZERO

SET WIND FLAG TO ZERO

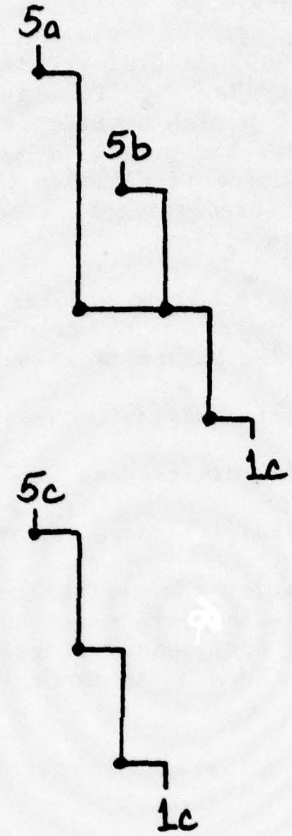
SAVE FORMER MASKS

SET MASKS FOR THE DIGITS

SET MAP FLAG TO ZERO

SAVE FORMER MASKS

SET MASKS FOR MISSED
APPROACH RANGE



VSPCL FLOWCHART (SHEET 5 OF 5)

Speech Understanding Subsystem

Speech understanding (SUS) on CPU1 receives recognized phrases from the SR module, SAID, over on CPU2. The IPB input receiver on CPU1 (IPBIN1) activates the SUS task. If a high confidence recognition is received, it is a simple matter to process the phrase as understood. However, complications arise when a low confidence recognition provides a second choice phrase or as misrecognitions or unrecognizable (too short or too long) phrases are encountered in a message stream.

The various speech understanding modules (refer to figure 20) attempt to concatenate phrase groups and to resolve low confidence conflicts and unrecognized phrases. The phrase types subject to concatenation include:

- headings and heading digits (subroutine SHEAD)
- wind, wind heading digits, and wind speed (subroutine SWIND)
- missed approach and its range (subroutine SMISH)

Stylization errors in which the trainee accidentally concatenates the first heading digit to the heading phrase are accounted for. This utterance will probably trigger a low confidence recognition for the heading followed by two correct digits. In such an instance the correctness of the first digit will be assumed.

Two stylization errors in which unnecessary pauses are inserted are also accounted for:

- X (pause) miles from touchdown (subroutine SDIGIT)
- Type of aircraft (pause) X (pause) X (pause) X,
  where XXX is the correct call sign (subroutine SMREC).

In the first case, a digit followed by a low confidence "miles from touchdown" triggers the association. A misrecognition accompanied by the correct call sign initiates the latter.

Low confidence conflicts which occur in headings, digits, and unsafe approach are resolved by selecting the correct choices relative to aircraft position at the time of the advisory's issuance. Another type of test is performed to investigate members of phrase groups (subroutine SMOTHR).

If a low confidence recognition is encountered, a check is made to see whether either of the choices qualifies as a trailer to the previously identified phrase. Messages of this nature are listed below. Commas demarcate phrases.

1. On the go, C/S, button X.
2. Climb and maintain 1500, turn right heading 300.

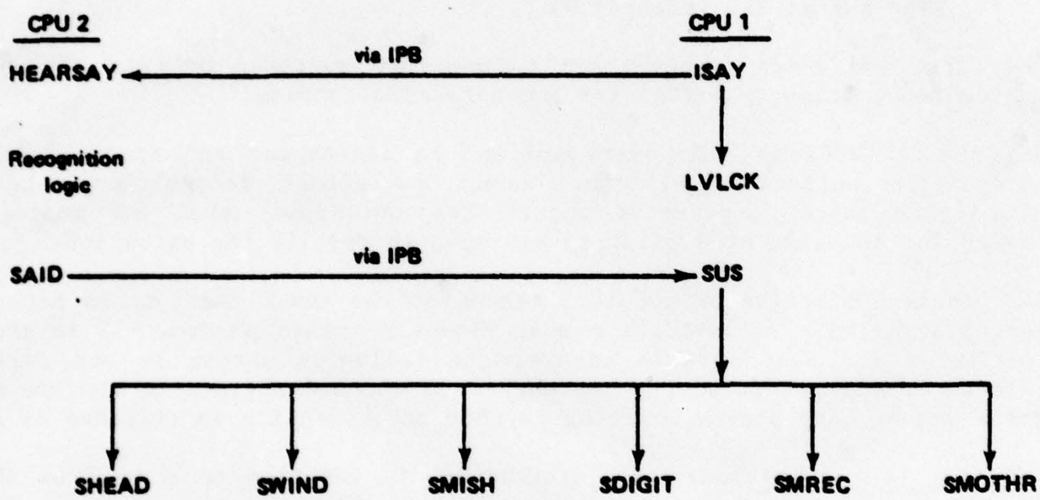


Figure 20. Speech Understanding Routines

3. C/S, missed approach, (map position), button X.

4. (course position), correcting.

5. At decision height, (unsafe approach condition), (roll out message).

6. Radar contact lost, (if runway not in sight), climb and maintain 3000, turn right, proceed direct point bravo hold until advised by GCA, over.

7. Climb and maintain 3000, turn right heading 270.

8. Wind XXX at XX, (clearance).

The first and second recognition choices are switched only if the second choice recognition qualifies for the antecedent phrase.

SUS module LVLCK is also ever vigilant in monitoring the speech level to detect fluctuations significant enough to affect recognition accuracy. After three such consecutive occurrences, an input level low message is issued for the simulated pilot to act upon to rectify the situation.

SUS modules, like the SR modules, depend on the reset that occurs after one second of silence following the most recently spoken phrase. It is assumed that if a one second pause occurs, the following phrase is not directly related to the previously spoken phrase in a message group or concatenation group sense. Any phrase awaiting further concatenation is released as is.

SUS Aids to Recognition. SUS activation, in addition to a call to SPEECH for SUSON, requires the activation of ISAY on CPU1.

ISAY saves model controller-generated messages to be sent to the recognition module HEARSAY on CPU2 upon the initiation of trainee voice input.

SUS

Description: Speech understanding controller processes speech recognition inputs which are shipped to it. Phrases which are understood are identified to APE and stored in the activity replay file.

Entry point: SUS

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN1

Cancelled by: N/A

Activates/calls: SHEAD, SWIND, SMISH, SDIGIT, SMREC, SMOTHR

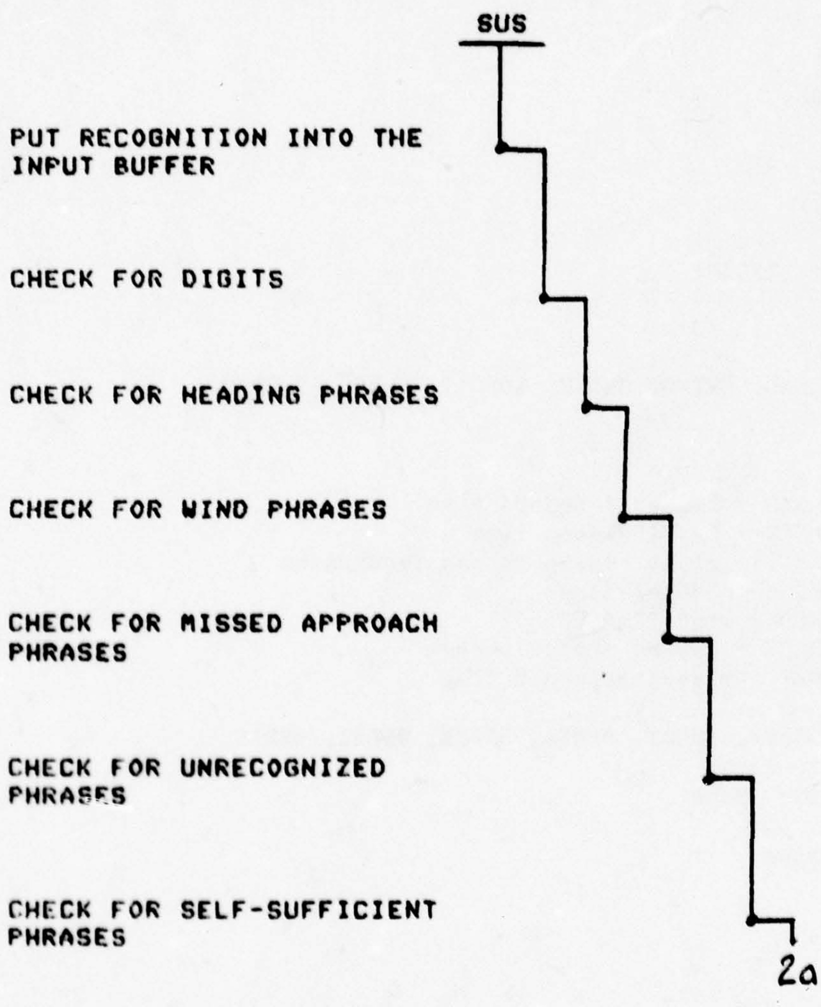
Cancels: None

Input arguments: IHSEC - LP_4 half second time
IMSEC - LP_4 100 msec time
IREC1 - first choice phrase recognized
IHDG - heading flag
IWND - wind flag
IREC2 - second choice phrase
IMAP - missed approach flag

Common variables: SSBFW, SSCAT, SSBFA, SSBFB, SSBFI, SSDIG

Files created/changed: None

Files referenced: None



SUS FLOWCHART (SHEET 1 OF 5)

IS THERE A PHRASE PENDING IN THE REPLAY BUFFER?

IS "CORRECTION" THE ONLY CHOICE IN "OTHER" BUFFER?

IS THERE A MESSAGE PENDING IN THE SPEECH UNDERSTOOD BUFFERS?

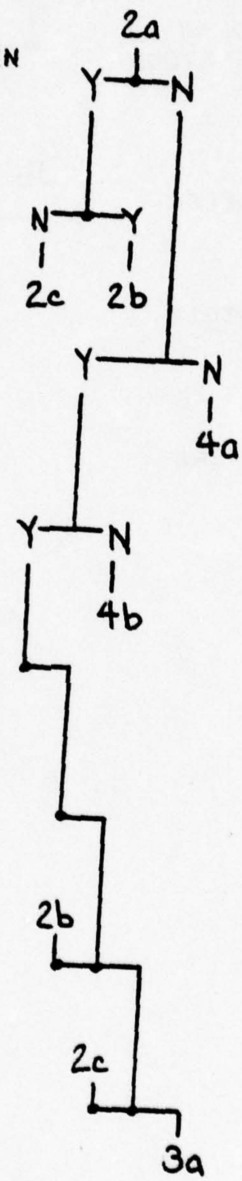
IS "CORRECTION" THE ONLY CHOICE IN "OTHER" BUFFER?

RELEASE MESSAGE IN SPEECH UNDERSTOOD BUFFER

PUT RELEASED MESSAGE IN REPLAY OUTPUT BUFFER

SET CORRECTION FLAG AND TIME IN REPLAY BUFFER

SEND OFF BUFFER TO REPLAY FILE



SUS FLOWCHART (SHEET 2 OF 5)

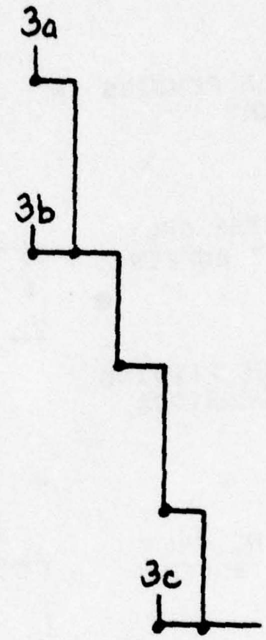
PUT "CORRECTION" INTO AN AVAILABLE SPEECH UNDERSTOOD BUFFER

RELEASE THE SPEECH BUFFER

RESET ALL FLAGS AND DIGIT BUFFERS

RESET ALL WORKING BUFFERS

DIE



SUS FLOWCHART (SHEET 3 OF 5)

INITIALIZE AN UNUSED BUFFER

IS THERE A HIGHEST RANKING MESSAGE FLAG BESIDES "OTHER"?

ARE THERE ANY CONFLICTS WITH WHAT IS ALREADY IN THE BUFFER?

FILL IN THE NEW INFORMATION

DOES THE MESSAGE ACCOUNT FOR ALL PHRASES RECEIVED?

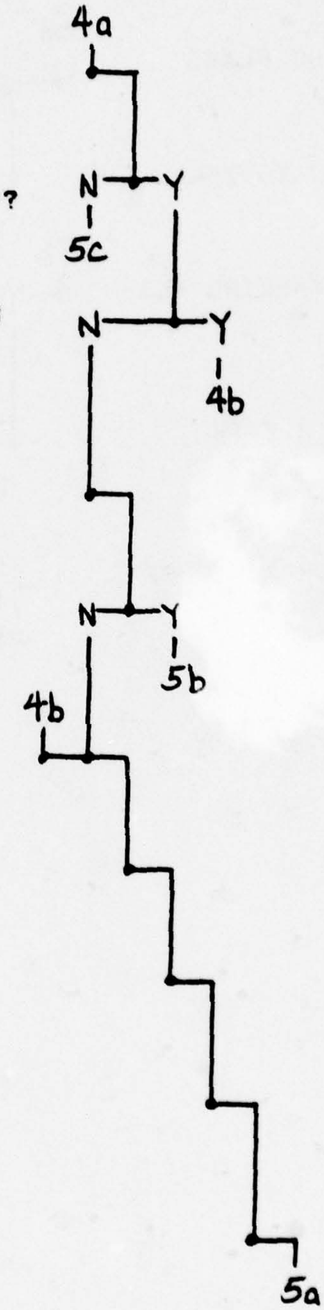
RELEASE THE BUFFER

FILL THE REPLAY BUFFER

SEND OFF THE REPLAY BUFFER

FIND AN AVAILABLE SPEECH BUFFER

FILL IT IN WITH THE "OTHER" MESSAGE



SUS FLOWCHART (SHEET 4 OF 5)

5a

RESET ALL MESSAGE FLAGS

SET "OTHER" FLAG TO ZERO

3c

5b

IS THE HIGHEST RANKING MESSAGE DONE? Y — N

3c

FILL THE REPLAY BUFFER

3b

5c

FILL AVAILABLE SPEECH BUFFER WITH "OTHER" MESSAGE

3b

SUC FLOWCHART (SHEET 5 OF 5)

SHEAD

Description: Processes heading messages for speech understanding.

Entry point: SHEAD

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: SSBFW(1,1:8), SSCAT(1), SSAZN, SSHDG, SSBFI

Files created/changed: None

Files referenced: None

SHEAD

IS THE HEADING MESSAGE FLAG SET?

SET THE HEADING MESSAGE FLAG TO THE HEADING COUNT

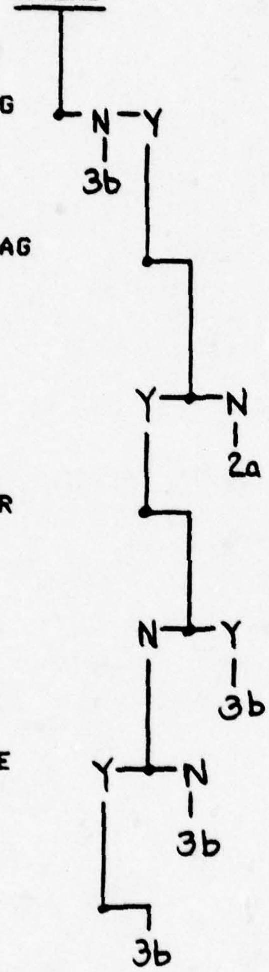
IS THIS THE HEADING?

FILL HEADING WORKING BUFFER

IS THE TURN IN THE APPROPRIATE DIRECTION?

IS THE SECOND CHOICE A MORE CORRECT TURN?

SWITCH FIRST AND SECOND CHOICES



SHEAD FLOWCHART (SHEET 1 OF 3)

FILL IN THE DIGITS:

DO WE HAVE THREE DIGITS?

ARE DIGITS IN AN APPROPRIATE DIRECTION FOR THE TURN?

IS THERE A SECOND CHOICE DIGIT THAT WOULD MAKE IT CORRECT?

USE THE SECOND CHOICE DIGIT

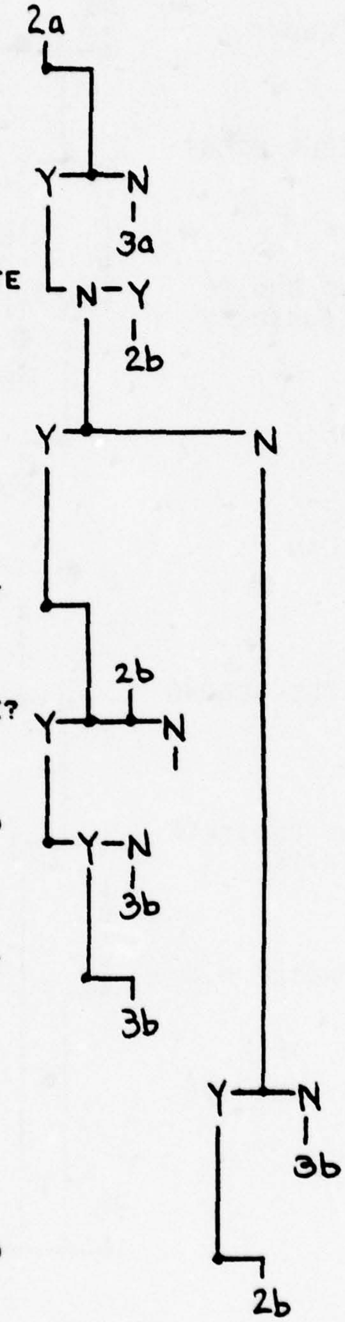
IS THE CORRECTION TOO LARGE?

IS THERE A DIGIT THAT WOULD MAKE IT CORRECT?

USE THE SECOND CHOICE DIGIT

IS THERE A SECOND CHOICE TURN THAT MATCHES THE HEADING?

SWITCH THE FIRST AND SECOND CHOICES



SHEAD FLOWCHART (SHEET 2 OF 3)

BRANCH ON DIGIT COUNT

IS THIS A PLAUSIBLE FIRST DIGIT?

IS THERE A SECOND CHOICE WHICH IS MORE PLAUSIBLE?

USE SECOND CHOICE

WAS THE HEADING LOW CONFIDENCE?

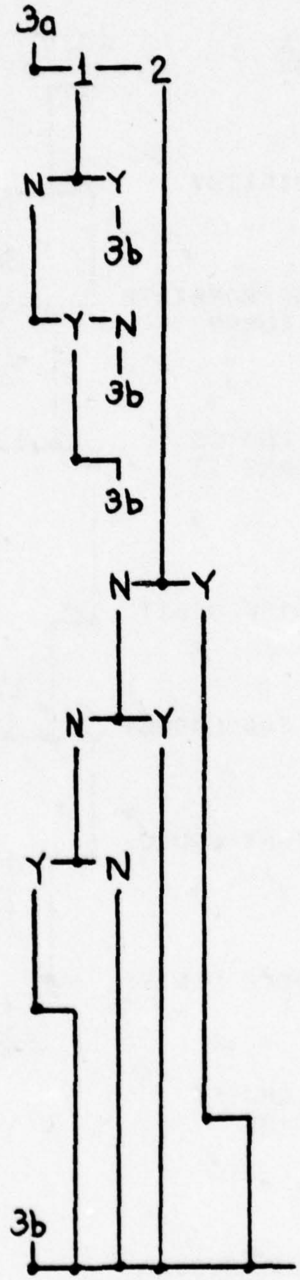
IS THIS A PLAUSIBLE SECOND DIGIT?

IS THERE A SECOND CHOICE WHICH WOULD BE BETTER?

USE THE SECOND CHOICE DIGIT

USE DIGITS AS IF LAST TWO DIGITS

RETURN



SHEAD FLOWCHART (SHEET 3 OF 3)

SWIND

Description: Processes wind messages for speech understanding.

Entry point: SWIND

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

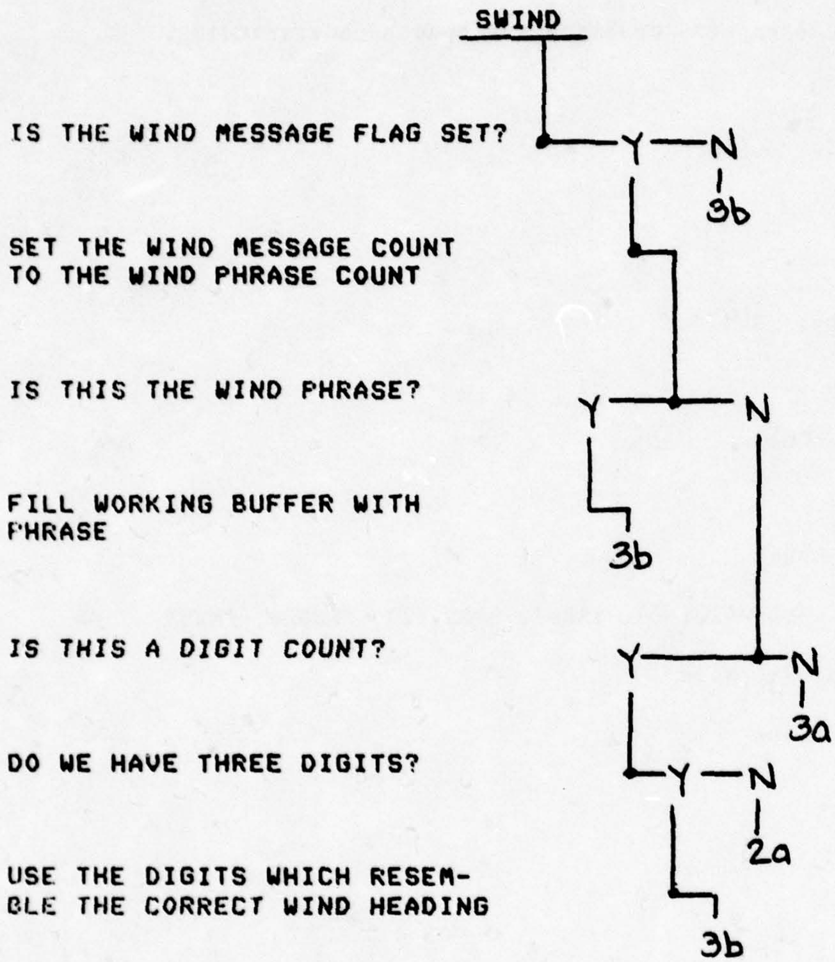
Cancels: None

Input arguments: None

Common variables: SSBFW(2,1:8), SSBFI, SSCAT(2), ENWHDG, ENWSP

Files created/changed: None

Files referenced: None



SWIND FLOWCHART (SHEET 1 OF 3)

BRANCH ON DIGIT COUNT

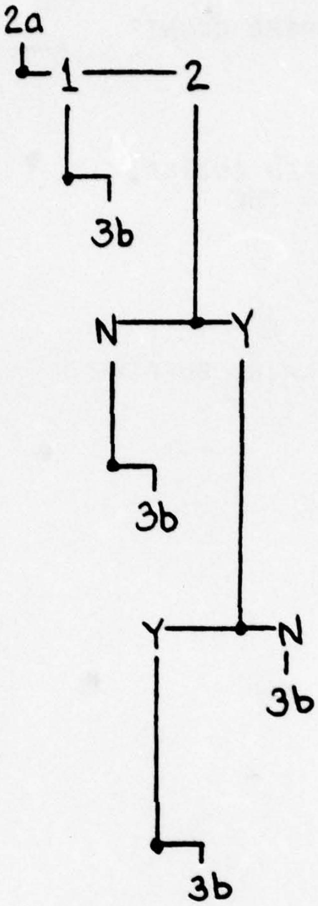
USE THE DIGIT THAT RESEMBLES
THE FIRST WIND HEADING DIGIT

WAS THE WIND MESSAGE LOW
CONFIDENCE?

USE THE DIGIT THAT RESEMBLES
THE SECOND WIND HEADING
DIGIT

IS THERE A COMBINATION OF
DIGIT CHOICES WHICH RESEMBLE
THE LAST TWO WIND HEADING
DIGITS?

USE THE DIGITS AS SUCH



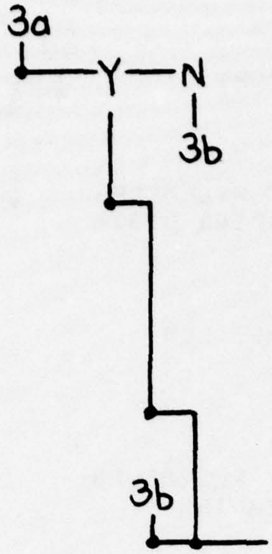
SWIND FLOWCHART (SHEET 2 OF 3)

IS THIS A WIND SPEED COUNT?

FIND THE WIND SPEED CHOICE
THAT IS CLOSER TO THE
CORRECT ONE

PUT IT IN THE WORKING BUFFER

RETURN



SWIND FLOWCHART (SHEET 3 OF 3)

SMISH

Description: Processes missed approach recognitions for speech understanding.

Entry point: SMISH

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

Cancels: None

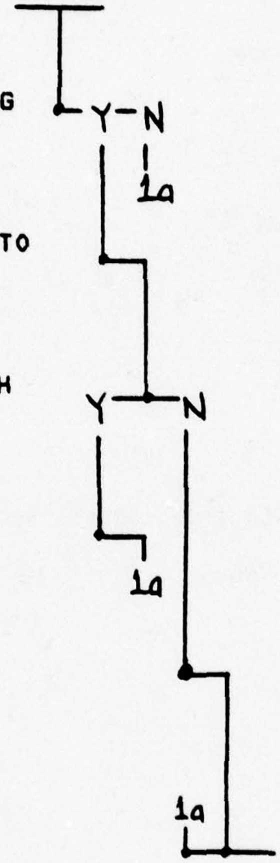
Input arguments: None

Common variables: SSBFW(3,1:8), SSBFI, SSCAT(3), ACRNG

Files created/changed: None

Files referenced: None

SMISH



IS THE MISSED APPROACH FLAG SET?

SET THE MAP MESSAGE COUNT TO THE MAP PHRASE COUNT

IS THIS THE MISSED APPROACH PHRASE?

FILL WORKING BUFFER WITH PHRASE

USE THE RECOGNITION CHOICE THAT RESEMBLES THE CORRECT MILE

RETURN

SMISH FLOWCHART (SHEET 1 OF 1)

SDIGIT

Description: Processes digits for speech understanding. Also checks for possible "miles from touchdown" advisory.

Entry point: SDIGIT

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: SSBFW(4,1:8), SSBFI, SSCAT(4), SSDIG, ACRNG

Files created/changed: None

Files referenced: None

IS THE DIGIT COUNT 3?

IS THERE A DIGIT RECOGNITION?

ARE THE WIND OR HEADING
FLAGS SET TO DIGIT COUNTS?

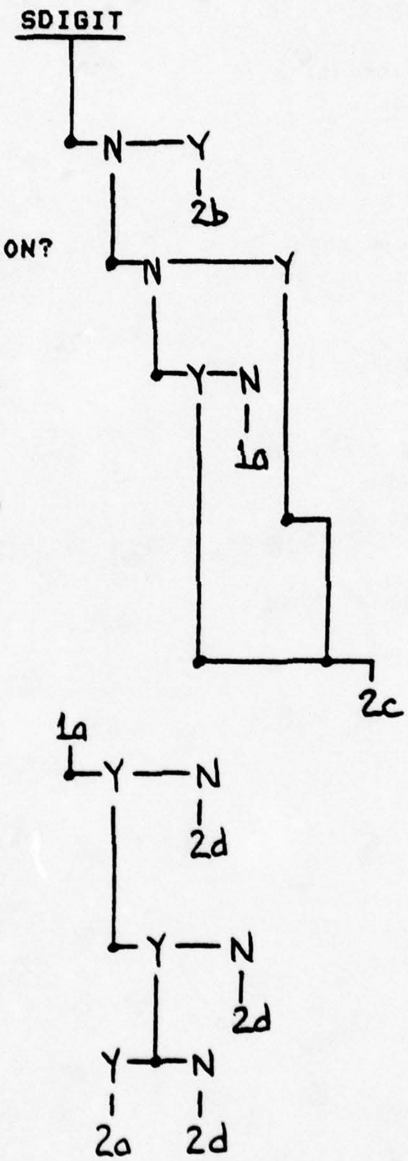
STORE THE FIRST AND SECOND
PLACE DIGIT

INCREMENT THE DIGIT FLAG

IS THIS A LOW CONFIDENCE
"X MILES FROM TOUCHDOWN"
MESSAGE OR MISUNDERSTOOD?

IS THERE ONE DIGIT STORED?

IS IT EQUAL TO THE ACTUAL
RADAR MILE?



SDIGIT FLOWCHART (SHEET 1 OF 2)

FORMULATE THE MILES FROM TOUCHDOWN MESSAGE IN THE WORKING BUFFER

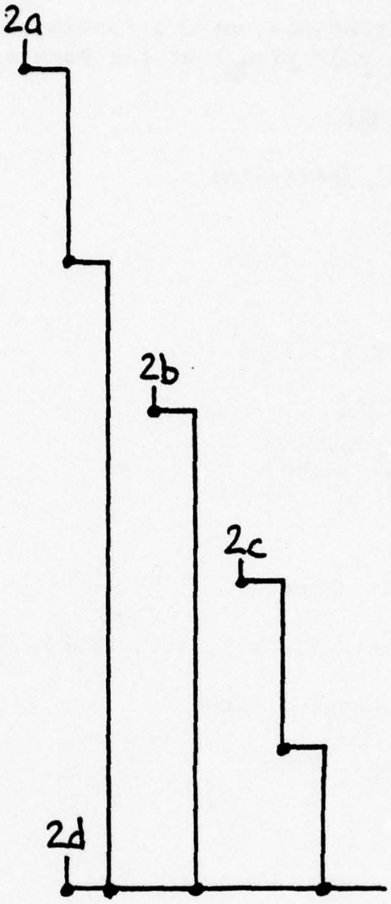
SET THE DIGIT FLAG TO ONE

UPDATE NUMBER IN WORKING BUFFER BY NEW DIGIT

FORMULATE A NUMBER USING DIGITS

PUT NUMBER IN WORKING BUFFER

RETURN



SDIGIT FLOWCHART (SHEET 2 OF 2).

SMREC

Description: Processes unrecognizable phrases. This routine looks out for a call sign that has been stylized by mistake.

Entry point: SMREC

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

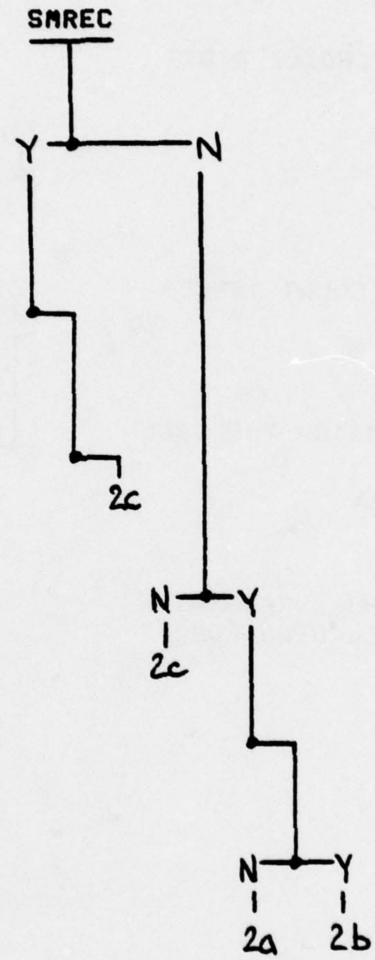
Cancels: None

Input arguments: None

Common variables: SSBFW(5,1:8), SSBFI, SSCAT(5), ACCS

Files created/changed: None

Files referenced: None



IS THE MESSAGE NOT UNDERSTOOD?

SET THE MESSAGE COUNT TO ZERO

FILL WORKING BUFFER WITH MESSAGE

IS THE PHRASE A DIGIT?

INCREMENT MESSAGE COUNT

DOES THE DIGIT MATCH THE CALL SIGN DIGIT?

SMREC FLOWCHART (SHEET 1 OF 2)

DOES THE SECOND CHOICE MATCH?

USE THE SECOND CHOICE DIGIT

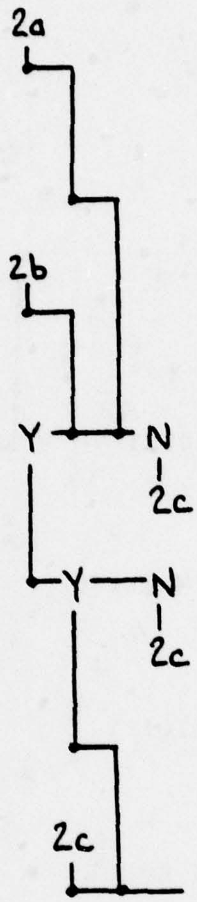
USE THE DIGIT

IS THE MESSAGE COUNT THREE?

DO THE DIGITS MATCH THE CALL SIGN?

FILL THE WORKING BUFFER WITH THE CORRECT CALL SIGN FORMAT

RETURN



SMREC FLOWCHART (SHEET 2 OF 2)

SMOTHR

Description: Processes all phrases. This routine looks out for low confidence phrases and tries to choose the best one based on a priori message information.

Entry point: SMOTHR

Classification: Subroutine

Period: None

Language: F

Activated/called by: SUS

Cancelled by: N/A

Activates/calls: None

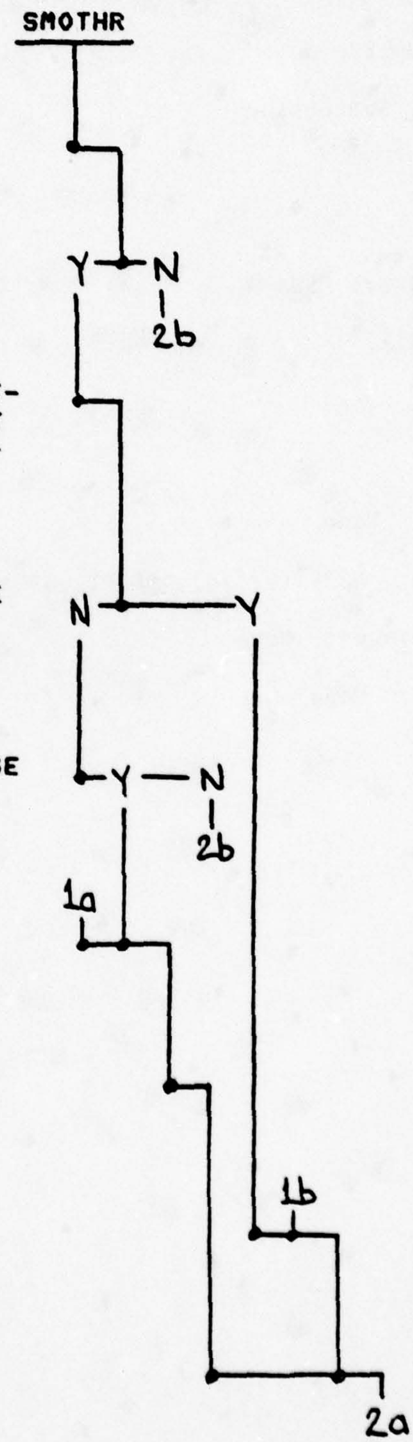
Cancels: None

Input arguments: None

Common variables: SSBFW(6,1:8), SSBFI, SSCAT(6), SSBFO, SSAZN

Files created/changed: None

Files referenced: None



INCREMENT "OTHER" FLAG

IS THIS A LOW CONFIDENCE MESSAGE?

SEARCH THE LIST OF PREDICTABLE PHRASES FOR THE FIRST AND SECOND CHOICE RECOGNITIONS

IS THE FIRST CHOICE PHRASE PREDICTABLE FROM THE LAST PHRASE UNDERSTOOD?

IS THE SECOND CHOICE PHRASE PREDICTABLE FROM THE LAST PHRASE UNDERSTOOD?

SWITCH FIRST AND SECOND CHOICES

PUT THEM IN THE WORKING BUFFER

PUT INPUT BUFFER INTO WORKING BUFFER

CLEAN UP WORKING BUFFER

SMOTHR FLOWCHART (SHEET 1 OF 2)

RETURN

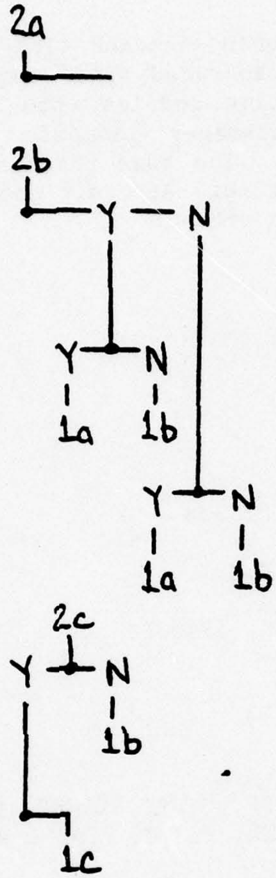
IS THE SECOND CHOICE PHRASE THE CORRECT CALL SIGN?

IS THIS THE FIRST PHRASE SAID IN THE LAST SECOND?

IS THERE CONFUSION BETWEEN TOO FAR RIGHT/LEFT FOR SAFE APPROACH?

IS THE RECOGNITION A CALL SIGN?

SET UP WORKING BUFFER WITH CALL SIGN IN IT



SMOTHR FLOWCHART (SHEET 2 OF 2)

ISAY

Description: Speech understanding task which ships performance measurement module-generated final controller message IDs to CPU2 speech recognition modules upon initiation of student voice input. These messages identify a valid set of final controller phrases. The same phrases are also sent out to the activity replay file. Aircraft position parameters of interest to SUS are also stored.

Entry point: ISAY

Classification: Task

Period: None

Language: F

Activated/called by: P23SUB

Cancelled by: PZ2, P3TRM

Activates/calls: LVLCK, IPBOUT1

Cancels: None

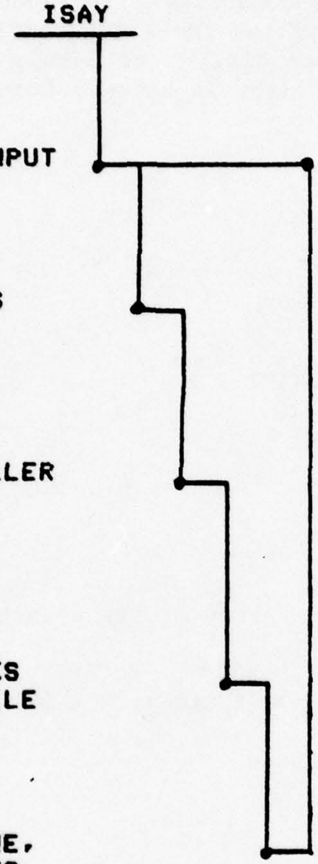
Input arguments: None

Common variables: CTGPP, CTGPT, CTCRP, CTCRT, CTRNG, CTOTHR, EVVST, SSHDG, SSAZN, SSRNG, ACAZN, ACRNG, ACHDG

Files created/changed: Activity replay file

Files referenced: None

ISAY



WAIT FOR STUDENT VOICE INPUT

ACTIVATE TASK WHICH WAITS
FOR END OF INPUT

SEND VALID SET OF CONTROLLER
PHRASES TO RECOGNITION
MODULES VIA IPB

OUTPUT SAME SET OF PHRASES
TO THE ACTIVITY REPLAY FILE

SAVE AIRCRAFT AZIMUTH ZONE,
RANGE, AND HEADING FOR SUS

ISAY FLOWCHART (SHEET 1 OF 1)

HEARSAY

Description: This recognition task receives controller messages from the IPB and applies the corresponding resolution masks to produce masks which single out likely phrases for recognition. The phase of flight is deduced from the messages.

Entry point: HEARSAY

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN2

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: IARGS - array of six possible final controller messages

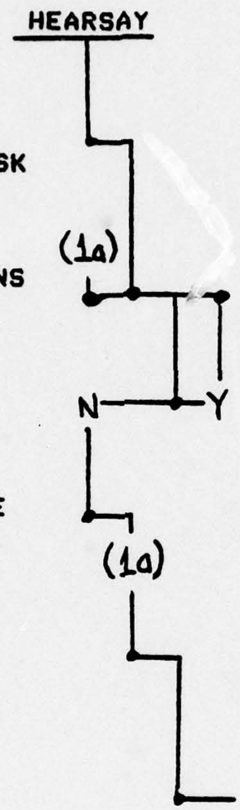
Common variables: RCGPP, RCGPT, RCCRP, RCCRT, RCRNG, RCOTHR, RCFZIS, RCBF(5), RCBF(6), RCBF(8)

Files created/changed: None

Files referenced: None

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

4 OF 9
AD
A069036



INITIALIZE SPECIAL RECOG-
NITION FLAGS AND PHASE MASK

LOOP FOR ALL SIX SELECTIONS

IS IT EQUAL TO -1?

"AND" SELECTION WITH PHASE
MASK

RESULT SHOULD BE A PHASE
INDICATION

DIE

HEARSAY FLOWCHART (SHEET 1 OF 1)

SAID

Description: Speech recognition task which ships recognitions over to CPU1.

Entry point: SAID

Classification: Task

Period: None

Language: F

Activated/called by: SUSON

Cancelled by: SUSOFF

Activates/calls: IPBOUT2

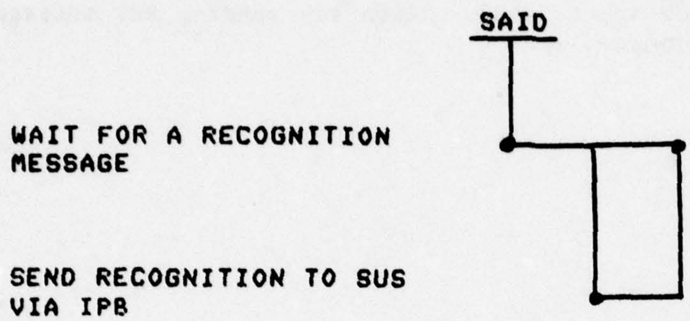
Cancels: None

Input arguments: None

Common variables: RCBF, BXCOG

Files created/changed: None

Files referenced: None



SAID FLOWCHART (SHEET 1 OF 1)

LVLCK

Description: This routine checks for low student voice input levels. Three consecutive low input levels cause a message buffer to be filled and an activity replay file record to be output concerning the low input level. Also any pending SUS messages are sent as complete.

Entry point: LVLCK

Classification: Task

Period: None

Language: F

Activated/called by: ISAY

Cancelled by: PHAZ2, PHAZ3, PRUN

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: SSBFA, SSBFB, SSCAT, EVVIN, SSDIG, SSBFW

Files created/changed: Activity replay file

Files referenced: None

INITIALIZE

WAIT FOR STUDENT VOICE INPUT TO CEASE

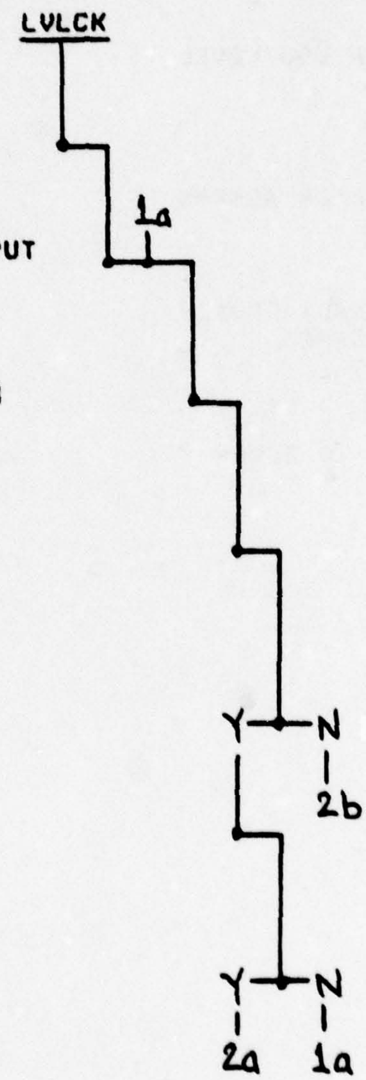
RELEASE ANY PENDING SPEECH BUFFER

RESET ALL SUS FLAGS AND TASK HOLDING BUFFERS AND DIGIT HOLDERS

IS INPUT LEVEL LOW?

INCREMENT INPUT LEVEL LOW COUNT

IS COUNT THREE?



LVLCK FLOWCHART (SHEET 1 OF 2)

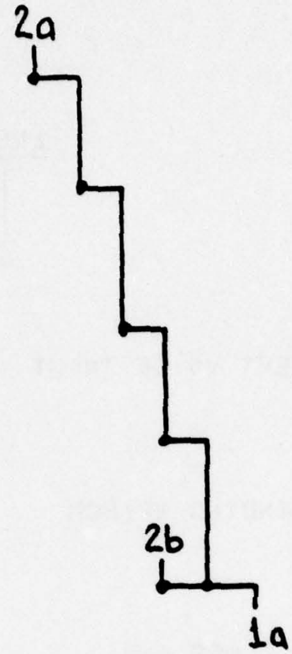
FIND AN UNUSED SPEECH BUFFER

FILL BUFFER WITH LOW LEVEL
MESSAGE

ALLOW SPEECH BUFFER ACCESS

SEND OUT "LOW INPUT LEVEL"
REPLAY FILE MESSAGE

RESET LOW COUNT TO ZERO



LVLCK FLOWCHART (SHEET 2 OF 2)

Simulations

The GCA-CTS provides a realistic environment in which the trainee can practice his skills. This is made possible by the simulations described in this section. They include simulations of the PAR environment and of other persons in that environment including an ideal GCA controller. In the paragraphs which follow, the aircraft, pilot and environmental simulation (APE) is discussed, followed by radar, display and finally model controller simulations. To ensure maximal transfer of training, face validity is the goal in all of these designs.

The following overview of the simulated environment will clarify the more detailed discussions which follow. First, the APE and radar simulations operate sequentially. Aircraft position updates are computed at the PAR sweep rate of .5 second. The timing problem is complicated somewhat by the fact that APE and the radar simulation operate in CPU1 while display processing takes place in CPU2. The scheme for maintaining the proper timing is shown in table 16. As shown, APE is queued at t_i to compute the environmental conditions and the aircraft position which will obtain at $t_{i+.5sec}$. As soon as these computations are complete, the radar simulation is called to compute the aircraft position in display coordinates. This display information is sent to CPU2 via the IPB. A .5 second periodic task in CPU2 displays the new aircraft position at the appropriate time.

The appropriate model controllers operate in parallel with APE and radar so that verbal responses are supplied and inputs to the simulated pilot are provided by the model controller during demonstrations.

These simulations speak by calling a speech output routine, GLIB which is also described in this section.

TABLE 16. SIMULATION TIMING

Time		Processing
t_i	–	APE begins its calculations for environmental conditions and aircraft position at $t_{i+.5sec}$
	–	Radar simulation is called to format aircraft position information for the display.
	–	Radar sends display information to CPU2 via the IPB.
	–	CPU2 receives the aircraft display data.
$t_{i+.5}$	–	CPU2 periodic task updates aircraft position information.

Aircraft/Pilot/Environmental Simulation (APE)

The purpose of APE, within GCA-CTS, is to provide simulated aircraft position information to the radar display module. This information determines the locations of the simulated targets on the display screen. The targets are required to move across the screen in a realistic manner and to respond realistically to controller advisories. Furthermore, such items as pilot personality and skill level, different aircraft types, random wind variation and various emergencies must be simulated.

An additional purpose of APE is to identify simulated pilot messages to the speech generation module so that the pseudo pilot can speak to the controller at the appropriate instant.

General Organization of APE. The following outline describes the organizational pattern.

A. Data Classes

We distinguish three classes of data related to APE and six modules which operate on these data. The data classes and modules and their relationship is illustrated in figure 21. The data classes are:

1. External Parameters:

   Values which are meaningful to the user (for the most part) and which are expressed in units convenient to him. These are values which control the nature of the simulated wind, aircraft, and pilot. Examples are mean wind speed in knots, heading of the runway in degrees (true), and magnetic variation.

2. Runtime Parameters:

   These are values which characterize the wind, aircraft, and pilot but which do not change ordinarily during the course of a simulation run. They are generally expressed in units convenient for use in the models such as feet, seconds, and radians. Runtime parameters are often strange combinations of external parameters concocted to make the runtime calculations as efficient as possible.

3. Runtime Variables:

   These are the detailed variables which change during a simulation. They include the APE simulation output, which is the current position of the aircraft, and many variables which are only of interest to the simulation models, such as the variance of the pilot's estimate of his height relative to the glideslope. Among the runtime variables are the state variables which are distinguished by the influence of their previous value on the current value being calculated (i.e., a state variable is "updated" by performing some operation on its former value).

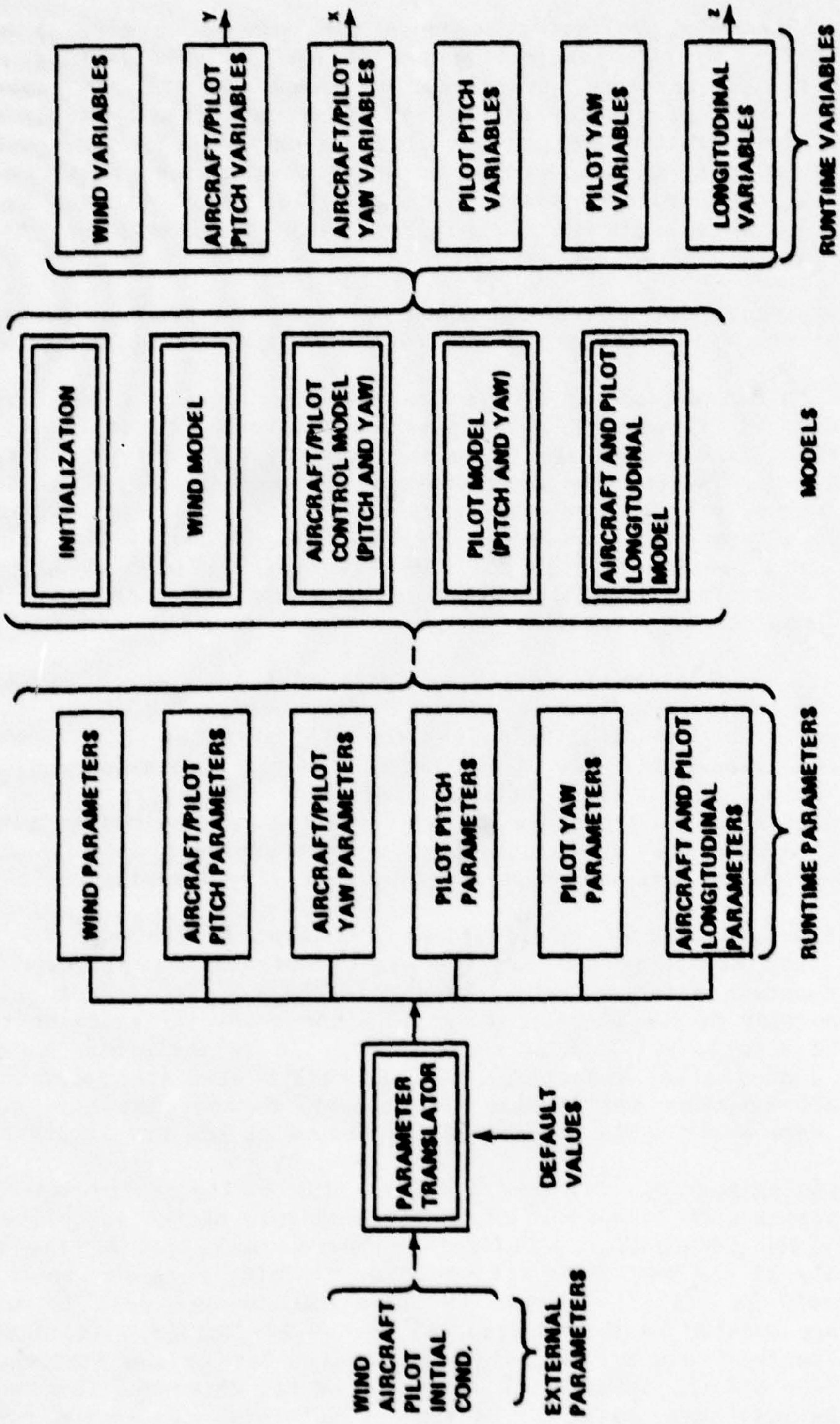


Figure 21. General Organization of APE

B. Processing Segments

Each of the six processing segments (or modules or models) has a unique relationship to the various data classes or subdivisions of the data classes. The parameter translator is unique in that it accepts external parameters as input and creates from those data (and possibly some default values stored internally) the runtime parameters. It is executed when the user wants to change the character of the simulation, normally only at the beginning of a training session or simulation run. (It may be possible to change parameters during a simulation, but no thought has been given to including that capability yet.)

The remaining five processing segments operate on the runtime parameters and current runtime variables to produce new values of runtime variables.

NOTE. The five processing segments operating during a simulation run will be described in terms of processing parameters and variables to produce new variables. The models are therefore entirely data driven and can be implemented in a natural way as re-entrant procedures, or at least procedures which can be executed an arbitrary number of times, depending upon how many "entities" are to be simulated. For example, if two "winds" are to be simulated it is necessary only to supply two sets of wind parameters and wind variables and execute initialization twice and the wind model twice. This holds true for two aircraft.

The five simulation processing segments fall into three categories determined by their rate of, or occasion for, processing. The initialization segment establishes initial values for all variables which need initialization and is executed once at the beginning of a simulation run (assuming one wind, pilot, and aircraft will be simulated. Otherwise various portions of the initialization procedure will be repeated as required to initialize each set of variables). The processing segments are best understood in terms of the modeling philosophy adopted which is briefly described below.

The pilot and aircraft combination is modeled in two separate parts. The first part simulates the way the pilot controls the aircraft to bring it into constant alignment with what he thinks is the correct path in space. This portion of the simulation reflects the characteristics of the aircraft, and the pilot's skill in controlling it. It is influenced by wind and updates (among other variables) the aircraft's simulated position in space. Aircraft dynamics and display requirements demand that this model be executed once every cycle (.5 second) to determine the new aircraft position.

The pilot's control of the aircraft is also influenced by what he thinks is the correct path in space. His opinion in this matter is influenced primarily by the information contained in advisories given by the student, and possibly by his own observations. For training purposes it is appropriate to model the pilot's opinion of where the correct path is as an initial estimate updated by the information in the advisories. An ideal pilot will make "perfect" use of the information given him by the trainee, whereas an imperfect pilot's opinion will wander, and his inference from new advisories will be less than perfect. In view of all this, the second portion of the

aircraft and pilot model simulates the evolution of the pilot's opinion of where the correct path is. It need only be called when he is forming or changing his opinion due to new information — generally only when an advisory is received.

This division of the aircraft/pilot simulation into two parts is appropriate for modeling the pitch and yaw motion of the aircraft, but a much simpler approach seems adequate for its longitudinal motion, as there is less variation of interest to the trainee in that dimension. Therefore, a single model, executed once every cycle, is proposed for longitudinal motion.

Finally, it appears quite feasible to use the same model to describe the pilot's control of the aircraft in both the pitch and the yaw plane, and also to use the same model to describe the evolution of his opinion of the vertical and of the horizontal position of the correct path in space. The parameters of his control and his inferential abilities in these two planes will be quite different, but the same set of equations describe his behavior. It therefore appears that an aircraft/pilot control model can be exercised twice each cycle, once to simulate control in the vertical plane and once for the horizontal plane. Similarly, a single pilot model can be exercised once each time he gets new glideslope information and once each time he gets new course information.

The result, then, is that the wind model and the aircraft and pilot longitudinal model must be executed once each cycle, the aircraft/pilot control model must be executed twice each cycle (once for pitch and once for yaw) and the pilot model must be executed when the pilot receives new information about the pitch plane or yaw plane position of the correct flight path.

Models. Models are described below for wind, longitudinal motion simulation, and the pitch part of aircraft/pilot control and pilot models. These models are presented primarily in their mathematical form, that is, optimization of the calculations is not complete, so these equations are more descriptive of the mathematical model of a physical or psychological process than of the final algorithm to be programmed.

Some background is required before these models are discussed. First, there is a resident random number generator accepting a seed value and providing a random integer between $-(2^{15}-1)$ and $+(2^{15}-1)$, inclusive, uniformly distributed over that interval and uncorrelated from sample to sample. In the following these random numbers are called RN, and the implication is that each time this symbol occurs a new random number is to be found and its value used. For example, the nonsense formula:

$$r = (RN+RN+RN)/32767$$

means that r is formed by summing three successive samples of the random number generator and dividing by 32767.

Secondly, the units of length, time, and angle internal to the simulation are feet, seconds, and radians.

Finally, the coordinate system adopted has its origin at the intersection of the glideslope with the ground, and its z axis horizontal and opposite to the direction of approach. The x and y axes are horizontal to the pilot's right, and vertical, respectively. A perfect approach is in the y-z plane, moving toward the origin, as illustrated in figure 22.

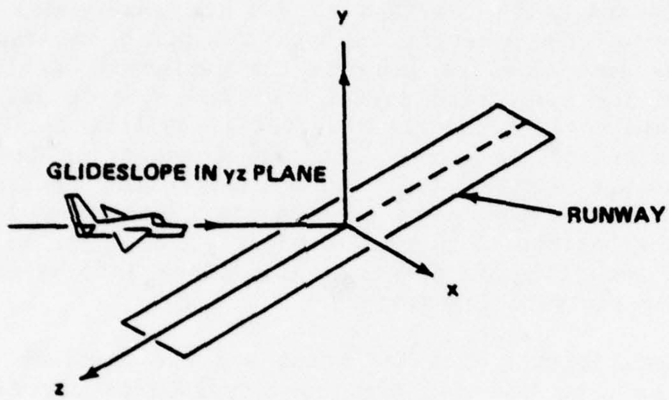


Figure 22. Coordinate System

A. Wind Model

Wind is modeled as the sum of a steady component and a random component, modified by gusts. The random component is modeled as the combination of two uncorrelated processes acting along and across the steady wind vector. Each of these processes has an auto covariance as a function of time. The result of this auto correlation is that successive samples of wind (each half second) are not independent, preventing wild variations, and the direction of the wind also varies less wildly.

Gustiness is modeled as occasional increases and decreases in the wind, affecting the component along and across the steady wind direction equally. Gusts and "antigusts" or decreases in wind intensity are assumed to occur equally frequently, with equal average durations. The steady wind speed is assumed to be equal to the geometric mean of the mean wind speed during gusts and antigusts along the steady wind direction. (So gusts of +100 percent intensity are accompanied by antigusts of -50 percent.)

During gusts, antigusts, and no gusts the wind variability is assumed to be twice as great along the steady wind component as across it.

The wind model is determined by the following external parameters:

H_R Runway heading. (True azimuth of the direction in which a plane lands) (degrees)

H_W Mean heading of the wind. (True azimuth of the direction from which the wind comes) (degrees)

S_{WN} Nominal windspeed. (Mean speed of the wind in the absence of gusts and along its mean heading) (knots)

V Wind variability parameter. If V=0 the wind is steady. If V=1 the standard deviation of the wind in its mean direction is 1/3 S_{WN}. The variability in the orthogonal direction is 1/2 as great; i.e., with V=1 the standard deviation is 1/6 S_{WN}. (dimensionless)

t_w Windspeed correlation time. (The time lag at which the auto-covariance of wind speed is 1/2 times its variance) (seconds)

S_{WG} Wind gust speed. (The mean wind speed in its mean direction during gusts) (knots)

F_G Fraction of the time gusts occur. (Gusts and antigusts occur equally often and for equally long periods of time, therefore, $0 \le F_G < 1/2$) (dimensionless)

T_G Mean duration of a wind gust or antigust (seconds). Must be larger than 0.5 second.

The runtime parameters used in the wind model are as follows:

$W = (H_R - H_W) \cdot \frac{\pi}{180} \bmod 2\pi$. Wind direction relative to z axis

$W_{zs} = 1.6887\ S_{WN} \cos\omega$. Nominal (steady) headwind

$W_{xs} = 1.6887\ S_{WN} \sin\omega$. Nominal crosswind

$k_1 = \cos\omega$

$k_2 = \sin\omega$

$\alpha = e^{-1/2t_w}$

$\beta = 1-\alpha$

$k_3 = \sqrt{\left(\frac{1+\alpha}{1-\alpha}\right)}\ \frac{v}{6} \cos\omega$

$k_4 = \sqrt{\left(\frac{1+\alpha}{1-\alpha}\right)}\ \frac{v}{6} \sin\omega$

$S_N = 1.6887\ S_{WN}$

$S_G = 1.6887\ S_{WG}$

$S_{AG} = S_N^{\ 2}/S_G$

$N_1 = 32767\ \left(\frac{F_G}{T_G(1-2F_G)} - 1 \right)$

$N_2 = 32767\ \left(\frac{2F_G}{T_G(1-2F_G)} - 1 \right)$

$N_3 = 32767\ \left(\frac{1}{T_G} - 1 \right)$

The model equations are as follows: They are executed once each half second.

1. $r = (RN+RN+RN)/32767$
2. $W_{1\ new} = \alpha W_{1\ old} + \beta r$
3. $r = (RN+RN+RN)/32767$
4. $W_{2\ new} = \alpha W_{2\ old} + \beta r$
5. $R = RN$
6. If state is not NOGUST then:

   i) If $R < N_3$ then: State = NOGUST
   $S_W = S_N$

   ii) go to 9
7. If $R < N_1$ then: State = GUST
   $S_W = S_G$
   go to 9
8. If $R < N_2$ then: State = ANTIGUST
   $S_W = S_{AG}$
   go to 9
9. $W_z = S_W (k_1 + 2k_3W_1 - k_4 W_2)$
   $W_x = S_W (k_2 + 2k_4W_1 + k_3 W_2)$

The output of the wind model is W_z and W_x, the instantaneous component of wind in the z and x directions. Steps 1 and 3 of this model form a pseudo-Gaussian random variable with zero mean and unit variance. It is limited to the interval [-3, +3] and has the distribution illustrated in figure 23. Steps 2 and 4 cause sample to sample correlation of the (otherwise uncorrelated) pseudo-Gaussian variables r. Steps 6, 7, and 8 implement a three-state Markov process which has states labeled NOGUST, GUST and ANTIGUST. The latter two states each occur fraction F_G of the time, with average duration T_G seconds at each occurrence. Step 9 produced the headwind and crosswind components of the wind.

The significance of some of the external parameters of the wind simulation is best understood through example. The simulation has therefore been exercised with two different sets of parameters and the results plotted, to illustrate the influence of the less obvious parameters.

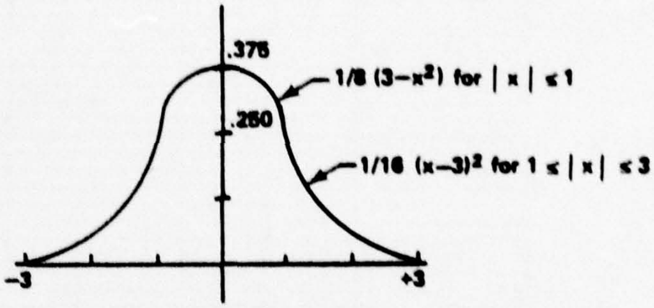


Figure 23. Probability Density of the Pseudo-Gaussian Random Variable Used to Form Wind Components.

Persons unfamiliar with the statistical treatment of sequences of numbers may find correlation time a new concept. Correlation time is a measure of how quickly a randomly varying value may change. If the wind speed is 5 knots greater than its average value at one instant, it will be almost 5 knots greater than its average value one microsecond later, showing that wind speed has a correlation time larger than one microsecond. In general, samples from a random process with correlation time t are very similar when the samples are taken at times differing by a small fraction of the correlation time, and are essentially independent when the time interval between samples is large compared to the correlation time. The effect is illustrated in figure 24. Thirty-one samples of two random processes are plotted in that figure. Each process has a mean value of zero, and an approximately Gaussian distribution of values. (The standard deviation of each process is indicated in the figure.) The first process is a sequence of entirely independent samples, so that successive values are uncorrelated, and the correlation time is zero. The second process (which was derived from the first by passing it through a first order digital filter) has a correlation time of six samples. Notice that when the second process takes on a low value, it tends to remain low, (etc.) and the sample-to-sample variation tends to be a smaller fraction of the standard deviation of the process than for an uncorrelated sequence.

The most pronounced effect of increasing correlation time is to reduce the frequency with which the process crosses its mean value. In thirty three samples of an uncorrelated process the average number of crossings will be sixteen, as indeed occurred in the sample of figure 24. The two samples of simulated wind plotted in figures 25 and 26 have correlation times of 8 and 0.5 seconds, respectively. Both the component along and across the average wind direction are plotted in these figures. The mean value of the cross component is zero and it is quite clear that the longer correlation time results in fewer zero crossings for that component. The same effect can be observed in the other component, although less clearly because the mean value differs in the gust, no-gust, and anti-gust states.

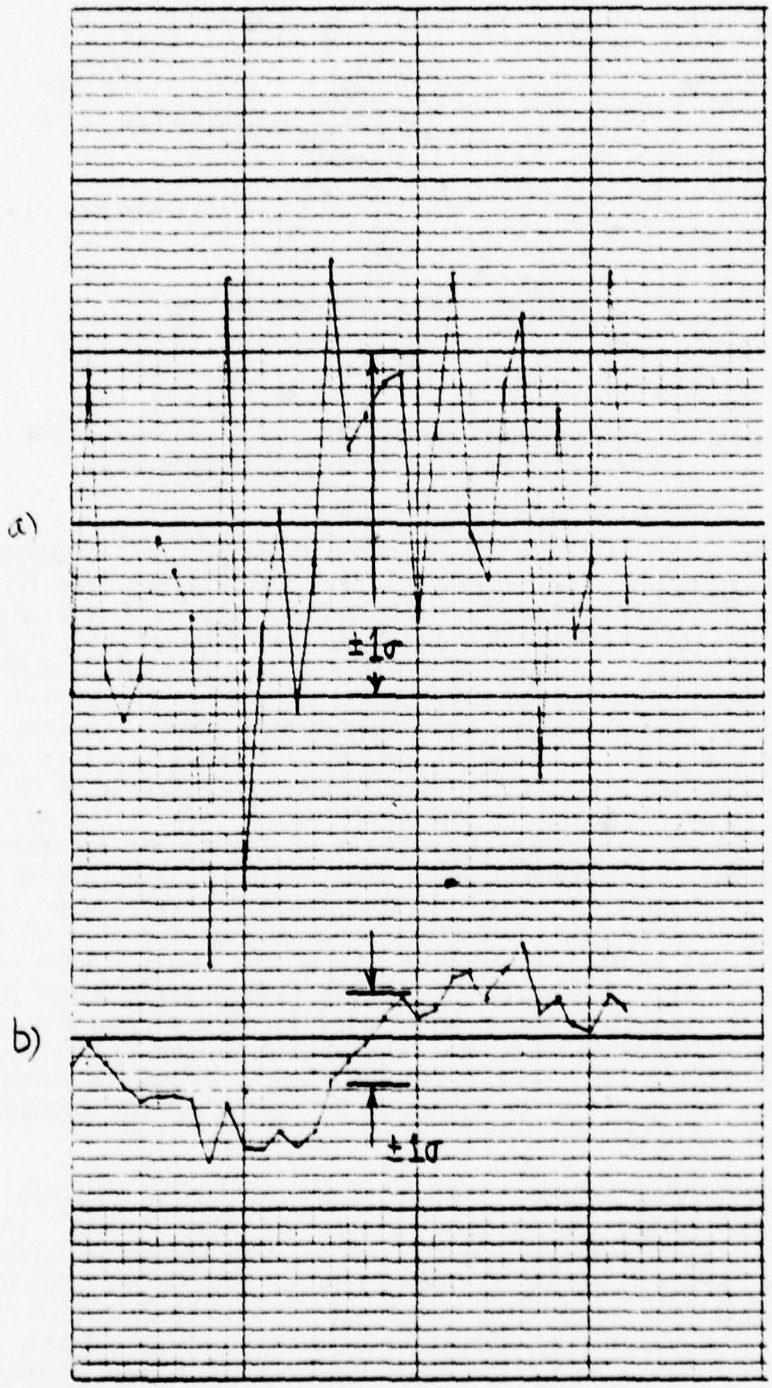


Figure 24. Random Sequences with Different Correlation Times
a) Correlation time t = 0. b) Correlation time t = 6 sample times

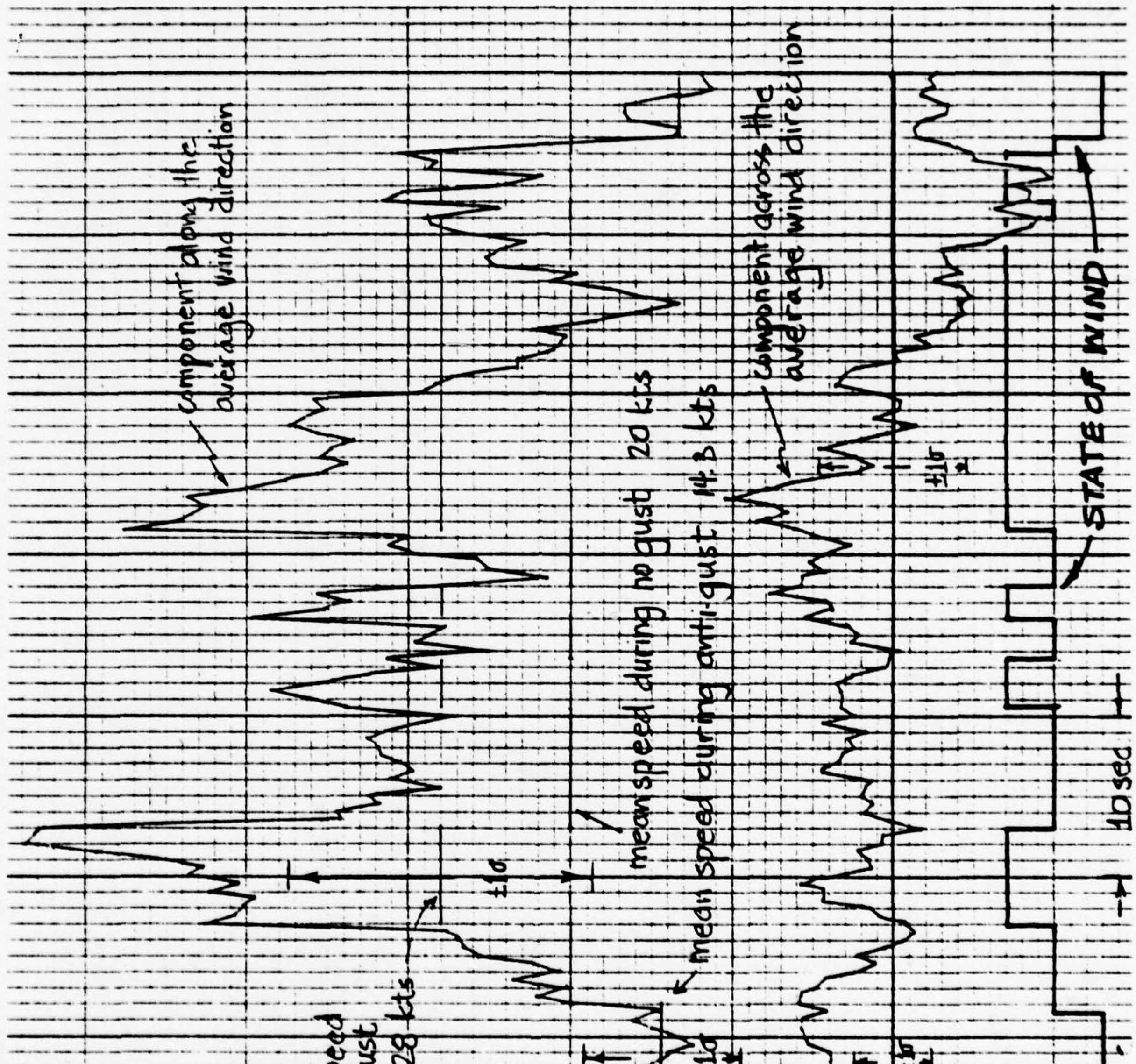


Figure 25. Sample of Simulated Wind With Correlation Time of 8 Seconds

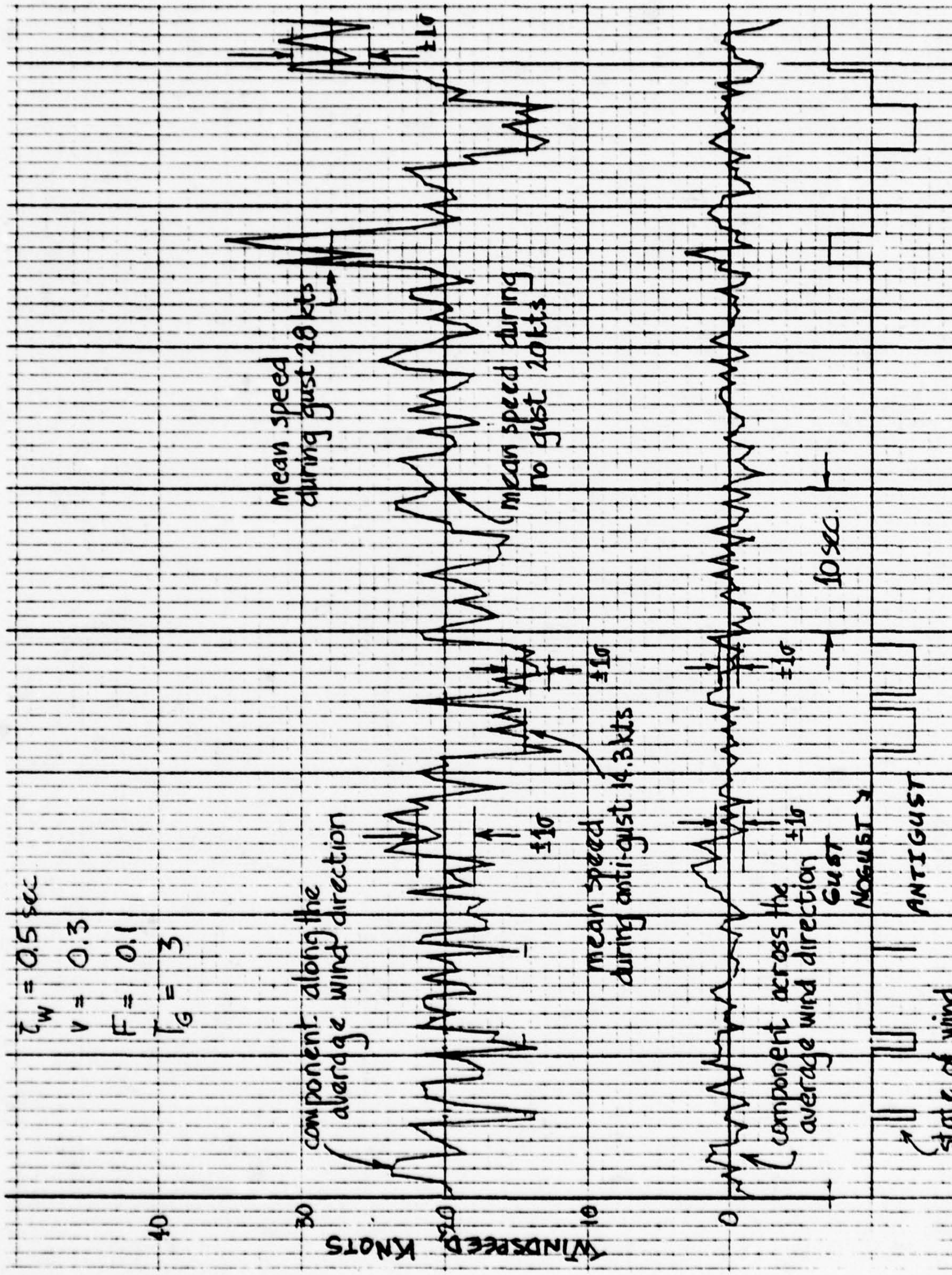


Figure 26. Sample of Simulated Wind with Correlation Time of .5 Second

The frequency of mean-value crossing is controlled entirely by the process correlation time and is not dependent on the process standard deviation. The large number of zero crossings of the cross component in figure 26 is thus properly attributed to the short correlation time (0.5 second) and not to the small standard deviation of that process. The standard deviation, on the other hand, is controlled by the variability parameter, V, and is entirely independent of the correlation time.

Another effect of correlation time on the wind simulation is to introduce sample-to-sample correlation of wind direction, as illustrated in figure 27. The tip of the two dimensional wind vector has been plotted in that figure, showing the variation of the wind for a period of about twenty seconds. The data were taken from the same simulation used to generate figures 25 and 26. Notice that during this interval the larger correlation time results in the wind dwelling to the left of its average direction.

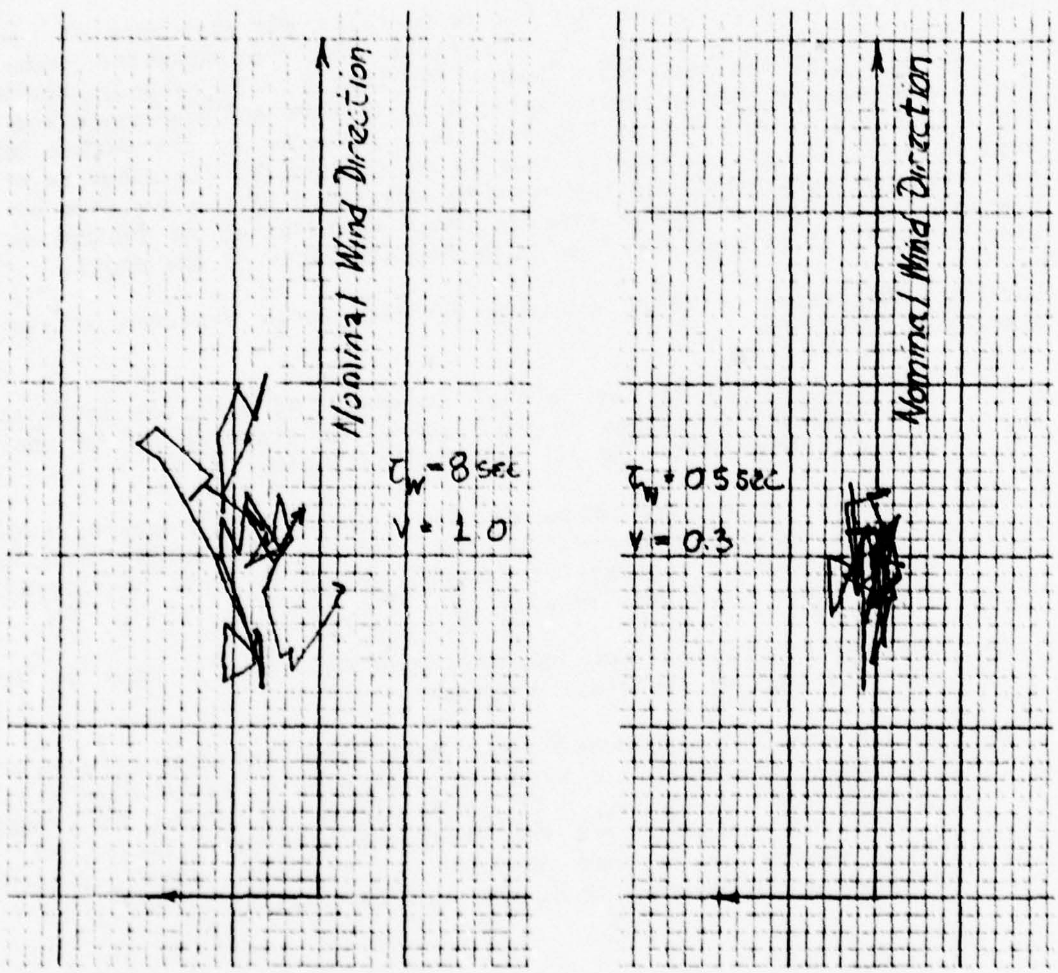


Figure 27. Two Dimensional Plots of Simulated Wind Samples

The influence of the variability parameter, V, is also apparent in these examples, as the standard deviation of the wind's variability about its mean value is 30 percent as great in the second example as in the first, resulting from V values of 1.0 and 0.3.

The state of the wind simulation (gust, no-gust, or anti-gust) is also plotted in figures 25 and 26 to illustrate the influence of the gust parameters. In the first example (figure 25) the parameter F_G is given a value of 0.25, indicating that, on the average, the wind should gust 25 percent of the time and anti-gust 25 percent of the time, leaving 50 percent of the time for "normal" windiness. The corresponding parameter value in the second example is 0.1, leading to a non-gusting condition 80 percent of the time. Notice that the wind states in these two examples are consistent with these average percentages, and yet neither the duration of any state nor the alternation between states is predictable.

Comparison of figures 25 and 26 also illustrates the influence of the mean gust duration parameter, T_G. In the first case the parameter value is 6 seconds and gusts and anti-gusts lasted for from 2 to 19 seconds, whereas in the second example the parameter value is 3 seconds and the gusts and anti-gusts lasted for from 0.5 to 3.5 seconds. In general, the Markov process used to model the gustiness will lead to an exponential distribution of gust and anti-gust durations, and the standard deviation of the duration of these states will therefore be the same as their mean value; an indication that wide variability in gust duration is a characteristic of the model.

Further examination of these examples verifies other features of the wind simulation, for example

- The variability of the "along" component of wind (as indicated by its standard deviation in each state) is proportional to the mean value in that state and the variability parameter, V.

- When the variability parameter, V, is unity, the standard deviation of the "along" component of the wind is one third its no-gust average value. (A negative "along" component would therefore be a three-sigma event when V=1.)

- The variability of the "across" component of the wind is always one-half that of the other component.

- When the difference between mean gust speed and mean no-gust speed ($S_{WG}-S_{NG}$) is small compared to the wind's variability in the no-gust state ($\sigma = VS_{NG}$) it is difficult to deduce the presence of gust or antigust from the windspeed history. The presence of these states is further obscured if the mean gust duration is similar to, or shorter than, the wind correlation time.

B. Aircraft/Pilot Control Model (Pitch component)

The pilot is assumed to have a notion of his current height above the glidepath and the slope of the glidepath. Let h be his assumed height above glidepath, and ξ the slope he imputes to it, as illustrated in figure 28.

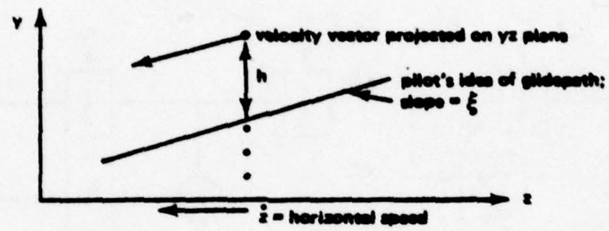


Figure 28. Pilot's Relationship to His Assumed Glidepath

Then the pilot controls his aircraft to bring it steadily on glidepath; i.e., to make both h and $\dot{h}$ identically zero. Using subscript 0 to indicate current conditions, and y to indicate the y coordinate of the aircraft, the pilot thinks his height above glidepath, as a function of time, is

$$h(t) = h_0 + \xi\ [z(t)-z_0] - y(t) + y_0$$

so the rate of change of his height error at the current instant is

$$\dot{h}_0 = \dot{y}_0 - \xi z_0$$

In trying to bring h and $\dot{h}$ to zero the pilot uses his kinesthetic sense of acceleration ($\ddot{y} = \ddot{h}$), and his personally and instrumentally observed sense of sink rate and altitude ($\dot{y}$ and $\ddot{y}$) to control the aircraft. In general, his ability to control the aircraft suggests a third order system. To correct pitch position he must 1) move the control surfaces to develop a change in angle of attack which causes a vertical component of acceleration 2) which causes a change in vertical component of velocity 3) which causes a change in vertical position. In pitch his control might be modeled nicely as a second order system because the pilot can produce the vertical acceleration quite quickly after deciding to do so. In yaw, however, he must cause the aircraft to roll before the desired lateral acceleration develops. The time constant of his ability to develop lateral acceleration can be large compared to the 0.5 second cycle time, for example for a heavily iced large airplane. The real reason for using a third order system to depict the pilot's control of the aircraft in pitch is that it is necessary in the yaw plane, and it is desirable to use the same model in pitch and yaw.

In the absence of disturbing influences such as wind and changes in the pilot's notion of where the glideslope really is, a linear and continuous model of his control of the aircraft is depicted in figure 29. We use $S_H = -\dot{z}$ to indicate the (positive) horizontal speed of the aircraft.

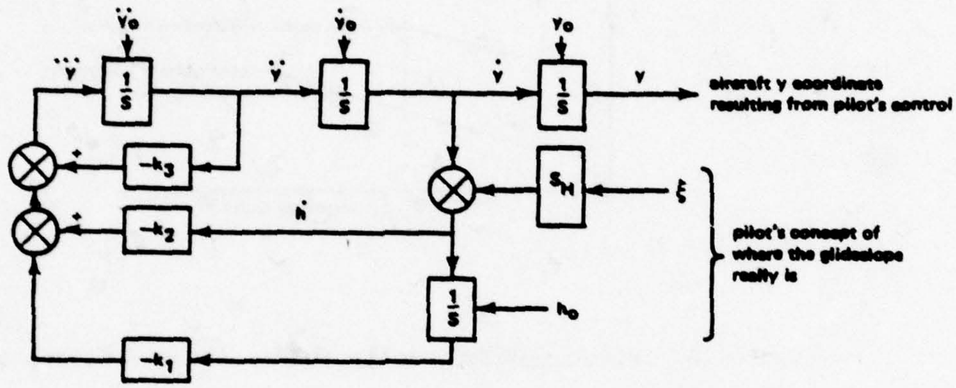


Figure 29. Linear Model of the Pilot's Control of His Aircraft

The three constants k_1, k_2, and k_3 in this model can be chosen to reproduce realistically the way a pilot of given skill level moves an aircraft of a given type in a given condition (iced or not iced) from an off-glidepath position to conform with his notion of what the glidepath is. Three independent parameters are both necessary and sufficient to determine k_1, k_2, and k_3. For example, the percentage overshoot, time to cross and rate of increase in vertical acceleration will suffice. A representative set of k's and a description of procedures for modifying them can be developed to produce any reasonable desired behavior of the pilot control model.

Modifications to the linear model. The linear model of the aircraft and the pilot's control of it to achieve a given glidepath must be modified in several respects to obtain more realistic behavior of the simulation. The model will be described further as if it were a continuous system, then it will be converted to the final digital form. The features which must be incorporated are:

- jerk limits
- pilot control asymmetry and acceleration avoidance
- wind effects

- ground avoidance
- pilot skill factors

Each of these modifications is discussed below:

1. Jerk Limits

As everyone who has ridden with an inconsiderate bus driver knows, the most disconcerting sensation in a vehicle may not be simple acceleration, but quick changes from acceleration to deceleration, or from one degree of acceleration to another. One characteristic of subjectively "smooth" motion is that acceleration changes don't occur more rapidly than a certain rate (the rate depends upon the axis in which they occur). The rate of acceleration is called "jerk" and to obtain realistic limits on the aircraft motion, jerk will be limited in absolute value to a small fraction of a g per second.

2. Pilot Control Asymmetry and Acceleration Avoidance

People also object to excessive acceleration, even when steadily applied. The acceleration exhibited by different aircraft will differ. A fighter pilot is accustomed to vertical turn accelerations of several g's when required. A pilot of a 707 with passengers aboard neither can nor would want to apply such vertical accelerations.

Furthermore, there is a decided asymmetry in the discomfort associated with upwards and downwards acceleration. Feeling forced out of the seat towards the canopy is decidedly uncomfortable for one not accustomed to it, and it is more or less avoided even by an experienced pilot, in comparison with upward acceleration which forces his posterior into the seat. Control asymmetry will be introduced as a non-linear element transforming the vertical acceleration indicated by the linear model (after jerk limits are imposed) into a limited value, such as depicted in figure 30.

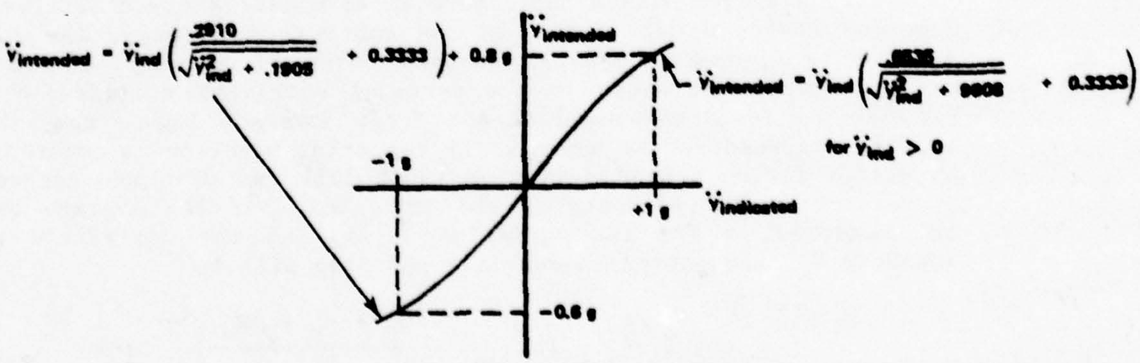


Figure 30. Nonlinear Transfer Function Relating Indicated Vertical Acceleration to Intended Vertical Acceleration

3. Wind Effects

The pilot adjusts the aircraft orientation to achieve a vertical acceleration consistent with his aims and constraints. The wind adds to this acceleration, due to its variability. Steady wind is not a source of trouble for the pilot as it is easy for him to control its effect. The variation in wind adds an unpredictable component of upward acceleration. Its effect can be determined with the help of figure 31.

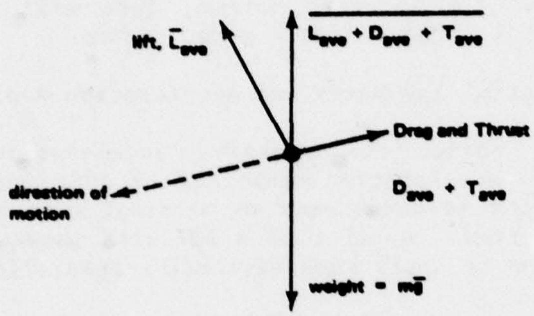


Figure 31. Vertical and Longitudinal Forces on an Airplane in Non-Turning State

Figure 31 shows the combined lift and drag and thrust experienced by the aircraft, averaged over a short term to eliminate the variability of the downwind component of wind. In the short term average, the lift and drag are compensated by the downward force of gravity, except for a residual component which is the vertical acceleration intended by the pilot. The instantaneous lift and drag, however, depend upon the aircraft's airspeed at the moment. If the pilot maintains a constant angle of attack during a wind deviation, both lift and drag are increased in proportion to the sequence of the windspeed. If the average speed of the headwind is W_H, its variation is W_H and the aircraft's groundspeed is S_G, the instantaneous lift and drag will be

$$\bar{L} + \bar{D} = (\bar{L}_{ave} + \bar{D}_{ave}) \left(\frac{S_G + W_H + \Delta W_H}{S_G + W_H} \right)^2$$

which leads to an unbalanced acceleration of magnitude

$$\Delta a = \left[\left(1 + \frac{\Delta W_H}{S_G + W_H}\right)^2 - 1\right] g \cong \left(1 + \frac{2\Delta W_H}{S_G + W_H}\right) g$$

which is largely in the vertical direction. Therefore wind is simulated by adding Δa to the vertical acceleration, $\ddot{y}$.

4. Ground Avoidance

Pilots are loath to drive their planes into the ground, and will try to avoid doing so even at the cost of not following the suggestion of a controller. An indication that the pilot should take action to avoid a crash is the constant vertical acceleration which, if applied continuously starting now, will bring him to ground with zero vertical velocity. If this acceleration exceeds some reasonable limit like one half or one g, then the required vertical acceleration should be applied immediately. The required acceleration is:

$$a_r = \frac{\dot{y}^2}{2y}$$

So y will be set to a_r whenever a_r exceeds a ground impact-avoidance acceleration limit, a_G.

5. Pilot Skill Factors

Perhaps the most important aspect of pilot skill in the controlled approach situation is his ability to infer where the glideslope really is from the cues and advisories available to him. That ability is modeled in the pilot model and is not discussed here.

A second aspect of pilot skill is his ability to correct perceived errors in position and path smoothly, quickly and without overcorrection. These characteristics of the modeled pilot's ability to control are determined by the gain factors k_1, k_2, and k_3. A third aspect of the pilot's ability is a random variation in the way he controls the aircraft, sometimes overcorrecting, sometimes undercorrecting. This feature of human fallibility is introduced into the aircraft/pilot control model in the form of varying gain at the control point ($\ddot{y}$). The gain is varied randomly, over an interval symmetric about unity. The gain is filtered to give it a pseudo-Gaussian distribution and a correlation time of 7.75 seconds.

A fourth aspect of pilot ability, related to his ability to correctly infer the position of the desired path in space from the clues available to him, is how clearly and precisely he is able to keep that path in mind as he tries to achieve it. A real pilot's notion of where the path is can be expected to deteriorate in time, in the absence of advisories and other correcting information. To model pilot "wander," a small random increment will be added to his estimate of his height above the glideslope and its slope, each cycle. His estimate of where the desired

path is will then exhibit random walk between corrections. The increments added are uniformly distributed, uncorrelated random variables, with a positive bias. Both the bias and the range of these random increments are proportional to pilot quality parameters. As a result the mean migration of the pilot's notion of where the glidepath is and what its slope is, will be proportional to the length of time since the last correction and the pilot quality parameters, and the standard deviation of the migration will be proportional to the square root of the time since the last correction and the pilot quality parameters. The pilot quality factors are given in terms of the mean drift rates, and the random component is determined so that after ten seconds without correction, the standard deviation of the drift is equal to two thirds the expected drift, assuring that the bias will become apparent fairly rapidly.

Final System Model. Incorporating the features just discussed into the original continuous linear system model of aircraft/pilot control given in figure 29 results in the system in figure 32.

The digital filter or discrete model can be developed as follows. The aircraft/pilot control model is influenced by the following external parameters:

k_1, k_2, k_3 — Linear system approximation coefficients characteristic of pilot and aircraft (to be described separately)

G_{JL} — Jerk limits (g/second). Pilot will not fly airplane with greater rate of change of acceleration than this.

G_{GA} — Ground avoidance acceleration (g). If pilot gets into a situation where crash can only be avoided by applying constant vertical acceleration of this amount or more, he does so.

PQ_1 — Standard deviation of random variations in pilot gain (dimensionless).

PQ_2 — Mean rate of drift of pilot's opinion of where the glidepath is in the absence of advisory data (feet per second, positive upward).

PQ_3 — Mean rate of drift of pilot's opinion of glidepath inclination in the absence of advisory data (degrees per second, positive moving the real glideslope towards horizontal).

Some runtime parameters are related to the external parameters as follows:

$$k_G = PQ_1\sqrt{93}$$

$$G_J = 32.2\ G_{JL}$$

$$G_G = 32.2\ G_{GA}$$

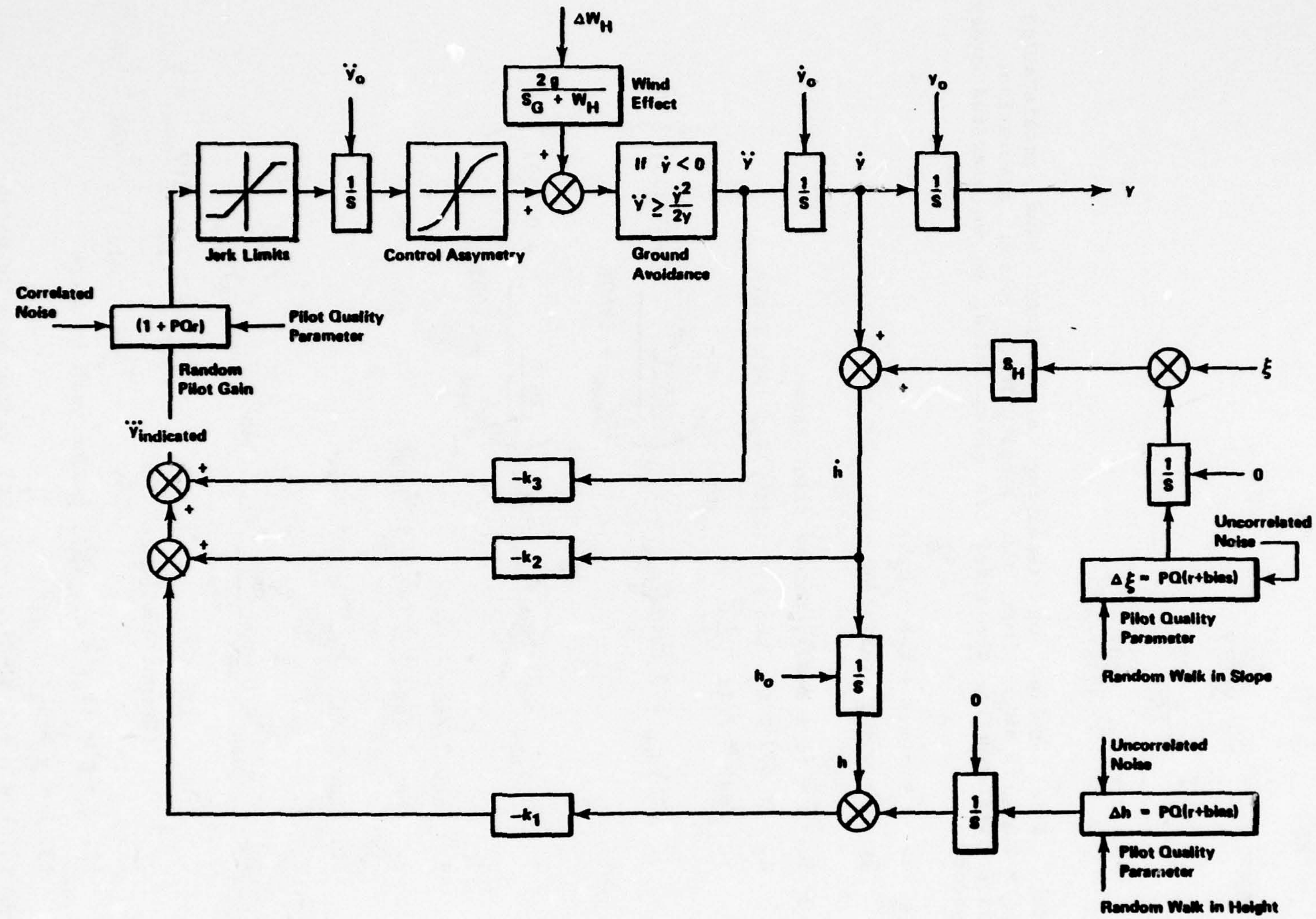


Figure 32. Continuous System Model of Aircraft/Pilot Control

$$r_{h1} = \frac{1}{2} PQ_2$$

$$r_{h2} = r_{h1} \frac{\sqrt{60}}{32767}$$

$$r_{s1} = \frac{1}{2} \frac{\pi}{180} PQ_3$$

$$r_{s2} = r_{s1} \frac{\sqrt{60}}{32767}$$

This model relies on receiving a vertical wind acceleration term $a_w = 2\Delta W_H/(S_G+W_H)$ from the longitudinal motion simulation. Equations implementing the model are given below, to be executed every half second.

1. $\dddot{y} = -(k_3\ddot{y} + k_2\dot{h} + k_1 h)$
2. $w = w + \frac{1}{16}(RN/32767 + w)$
3. $\dddot{y} = (1 + k_G w)\dddot{y}$, random pilot gain
4. If $|\dddot{y}| > G_J$ then $y = \text{sign}(\dddot{y})G_J$, jerk limit
5. $\ddot{y}_{new} = \ddot{y}_{old} + \frac{1}{2}\dddot{y}$
6. If $\ddot{y}_{new} > 0$ then $\ddot{y}_{new} = \ddot{y}_{new}\left(\frac{.6535}{\sqrt{\ddot{y}_{new}^2 + .9608}} + 0.3333\right)$
7. If $\ddot{y}_{new} < 0$ then $\ddot{y}_{new} = \ddot{y}_{new}\left(\frac{.2910}{\sqrt{\ddot{y}_{new}^2 + .1905}} + 0.3333\right)$
8. $\ddot{y}_{new} = \ddot{y}_{new} + a_w$
9. $\dot{y}_{new} = \dot{y}_{old} + \frac{1}{4}(\ddot{y}_{old} + \ddot{y}_{new})$
10. $y_{new} = y_{old} + \frac{1}{4}(\dot{y}_{old} + \dot{y}_{new})$
11. If $\dot{y}_{new} < 0$ and $\frac{\dot{y}_{new}^2}{2y_{new}} \geq G_A$ then $\ddot{y}_{new} = \frac{\dot{y}_{new}^2}{2y_{new}}$

    (Ground Avoidance)

    $\dot{y}_{new} = \dot{y}_{old} + \frac{1}{4}(\ddot{y}_{new} + \ddot{y}_{old})$

    $y_{new} = y_{old} + \frac{1}{4}(\dot{y}_{old} + \dot{y}_{new})$
12. $\xi = \xi - (r_{s1} + r_{s2} RN)$, random walk in slope
13. $\dot{h} = \dot{y} + S_H\xi$
14. $h = h + \frac{1}{2}\dot{h} - (r_{h1} + r_{h2}RN)$, random walk in height
15. $\ddot{y}_{old} = \ddot{y}_{new}$

    $\dot{y}_{old} = \dot{y}_{new}$

    $y_{old} = y_{new}$

C. Pilot Model (Pitch Component)

The pilot pitch model is developed in the following paragraphs by considering what happens to a "perfect" pilot who uses occasional cues to improve his estimate of where the glideslope really is. A model of this behavior is developed in the form of a Kalman filter.

Pilot Estimates. Suppose at time t_0 the pilot thinks he is at altitude h above the glideslope, and the glideslope has slope ξ. These data, together with his position, determine where in the pitch plane he thinks the glideslope really is. Suppose further that he is in error by amounts Δh and $\Delta \xi$, and that the true position of the glideslope is characterized by h^* and ξ^*. Then

$$h = h^* + \Delta h$$
$$\xi = \xi^* + \Delta \xi$$

The pilot will think his estimates are without bias (else he would change them) and they are in varying degrees of quality or definiteness. This notion is quantified as the variance and covariance of the errors Δh and $\Delta \xi$. Using $\langle\ \rangle$ to indicate expected values, suppose that

$$\langle \Delta h^2 \rangle = \sigma_{11} \text{ the variance of } \Delta h$$
$$\langle \Delta h \Delta \xi \rangle = \sigma_{12} \text{ the covariance of } \Delta \xi \text{ and } \Delta h$$

and

$$\langle \Delta \xi^2 \rangle = \sigma_{22} \text{ the variance of } \Delta \xi.$$

After some time has passed, the pilot's conception of his position relative to the glideslope will change. At time $t_1 > t_0$ he will think he is at weight h_1 over the glideslope and that its slope is ξ_1. If he has no advisories or other clues which would change his mind as to the location of the glideslope, he now thinks the glideslope is at

$$h_1 = h_0 + h_w + (y(t_0) - y(t_1)) + (\xi + \xi_w)\, dz$$
$$\xi_1 = \xi + \xi_w$$

where

$y(t_0) - y(t_1)$ is the altitude lost
dz is the distance traveled in the z direction

and

h_w, ξ_w are changes in his opinion of where the glideslope is due to "wander," or his inability to keep the slope firmly in mind.

The new errors in his opinion of h and ξ are therefore

$$\Delta h^{new} = \Delta h - h_w - (\Delta\xi + \xi_w)\, dz$$
$$\Delta\xi^{new} = \Delta\xi - \xi_w$$

Again we ignore the bias in his "wander," as he is entirely unaware of it, else he would remove it. His confidence in his new estimates, h_1 and ξ_1, will be different from what it was at t_0, partly because he wasn't quite sure what the slope was (and he has covered distance dz) and partly because his concept of where the path is is becoming fuzzier due to wander. These facts are revealed in the new variances and covariances: notice if the wander terms are assumed to be statistically independent of the initial height and slope errors, then

$$\sigma_{11}^{new} = \langle \Delta h^{new2} \rangle = \sigma_{11} - 2\sigma_{12} dz + \sigma_{22} dz^2 + \langle h_w^2 \rangle - 2\langle h_w \xi_w \rangle dz + \langle \xi_w^2 \rangle dz^2$$
$$\sigma_{12}^{new} = \langle \Delta h^{new} \Delta\xi^{new} \rangle = \sigma_{12} - \sigma_{22} dz + \langle h_w \xi_w \rangle + \langle \xi_w^2 \rangle dz$$
$$\sigma_{22}^{new} = \langle \Delta\xi^{new2} \rangle = \sigma_{22} + \langle \xi_w^2 \rangle$$

If it is further assumed that the pilot's wander in h and ξ are independent, the crosscorrelation term $\langle h_w\ \xi_w \rangle$ vanishes, further simplifying the first two equations. We have modeled "wander" as independent random walks. That is, in $(t_1 - t_0)$ seconds, the accumulated "wander," h_w, is the sum of $2(t_1 - t_0)$ independent random samples, each with standard deviation σ_{wh}, and similarly for wander in ξ. Therefore

$$\langle h_w^2 \rangle = n\sigma_{wh}^2$$

and $\quad \langle \xi_w^2 \rangle = n\sigma_{w\xi}^2$

where $n = 2(t_1 - t_0)$, the number of intervening cycles

so

$$\sigma_{11}^{new} = \sigma_{11} - 2\sigma_{12} dz + \sigma_{22} dz^2 + n(\sigma_{wh}^2 + \sigma_{w\xi}^2 dz^2)$$
$$\sigma_{12}^{new} = \sigma_{12} - \sigma_{22} dz + n\sigma_{w\xi}^2 dz$$
$$\sigma_{22}^{new} = \sigma_{22} + n\sigma_{wh}^2$$

Assume now that an advisory arrives at t_1, or some other event occurs which gives the pilot new data about where the glideslope really is. In general these data are to be treated as an estimate of h and ξ, to be combined in some way with the pilot's previous opinion of where the glideslope is. Being an optimal pilot, he will combine these data in an optimal way. A good model for the result is a minimum-variance estimate, or

Kalman filter. It is developed as follows. Let the new data imply a height and slope h_a, ξ_a. Suppose that (the pilot thinks that) the variances of these estimates are σ_{ha}^2 and σ_a^2, and that they are without bias and uncorrelated. He will then change his estimate of ξ and h to reflect the new data by adding to his old estimate a fraction of the difference between the new estimate and his previous one, using a matrix weighting factor:

$$h^{new} = h_1 + W_{11}(h_a - h_1) + W_{12}(\xi_a - \xi_1)$$

$$\xi^{new} = \xi_1 + W_{21}(h_a - h_1) + W_{22}(\xi_a - \xi_1)$$

The matrix of weights,

$$W = \begin{vmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \end{vmatrix}$$

is uniquely determined by the condition that the variance of the new estimates should be as small as possible. The result is

$$W = \begin{vmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{vmatrix}_{new} \left(\begin{vmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{vmatrix}_{new} + \begin{vmatrix} \sigma_{ha}^2 & 0 \\ 0 & \sigma_{\xi a}^2 \end{vmatrix} \right)^{-1}$$

and furthermore, that the matrix of variances and covariances of this new, better estimate is given by:

$$C = (I-W) \begin{vmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{vmatrix}_{new} (I-W)^T + W \begin{vmatrix} \sigma_{ha}^2 & 0 \\ 0 & \sigma_{\xi a}^2 \end{vmatrix} W^T$$

(These are the covariance data needed to start the process again.) Surprisingly, this all boils down to a computationally feasible procedure. The results obtained so far are summarized below as a computational procedure, before consideration is given to filling in some practical details.

Presume that at time t_0 the following data are known:

h	the pilot's h estimate
ξ	the pilot's ξ estimate
$\left.\begin{matrix}\sigma_{11}\\ \sigma_{12}\\ \sigma_{22}\end{matrix}\right\}$	variances and covariance of h and ξ

and that at time t_1 new data are obtained indicating that

$$\left.\begin{array}{l} h_a \\ \xi_a \end{array}\right\} \text{ are the real h and } \xi \text{ values, and}$$

$$\left.\begin{array}{l} V_{ha} \\ V_{\xi a} \end{array}\right\} \text{ are variances of the new } h_s \text{ and } \xi \text{ estimates. (Previously called } \sigma_{ha}^2, \sigma_{\xi a}^2\text{)}$$

then the pilot updates both his estimate of h and ξ and their variances and covariances as follows:

dz = distance traveled in z direction

n = $2(t_1-t_0)$ the number of cycles which have passed

V_{wh} = variance of single step wander in h (previously σ_{wh}^2)

V_w = variance of single step wander in ξ (previously $\sigma_{w\xi}^2$)

calculate as follows:

$$
\begin{aligned}
V_{wh}' &= \mu V_{wh} \\
V_{w\xi}' &= \mu V_{w\xi} \\
\sigma_{11} &= \sigma_{11} + V'_{wh} - dz\,(2\sigma_{12} - dz(\sigma_{22} + V'_{w\xi})) \\
\sigma_{12} &= \sigma_{12} - dz(\sigma_{22} - V'_{wh}) \\
\sigma_{22} &= \sigma_{22} + V'_{w\xi} \\
S_1 &= \sigma_{11} + V_{ha} \\
S_2 &= \sigma_{22} + V_{\xi a} \\
\Delta &= S_1 S_2 - \sigma_{12}^2 \\
V'_{ha} &= V_{ha}/\Delta \\
V'_{\xi a} &= V_{\xi a}/\Delta \\
W_{11} &= S_2\, V'_{ha} \\
W_{12} &= \sigma_{12}\, V'_{\xi a} \\
W_{22} &= S_1\, V'_{\xi a} \\
\Delta h &= h - ha \\
\Delta\xi &= \xi - \xi a \\
h &= h_a + W_{11}\,\Delta h + W_{12}\,\Delta\xi \\
\xi &= \xi_a + W_{12}\,\Delta h + W_{22}\,\Delta\xi \\
\sigma_{11} &= W_{11}\,\sigma_{11} + W_{12}\,\sigma_{12} \\
\sigma_{12} &= W_{11}\,\sigma_{12} + W_{12}\,\sigma_{22} \\
\sigma_{22} &= W_{12}\,\sigma_{12} + W_{22}\,\sigma_{22}
\end{aligned}
$$

This calculation requires 22 multiplication-like operations and 20 addition-like operations, which seems very reasonable since they would only be performed when the pilot received new information, which probably averages once every 4 to 5 seconds.

The Kalman filter approach to modeling the pilot's subjective inference of desired path location has several advantages. One is that (as described above) it will probably outperform any real pilot if implemented without disturbing influences. As it is a minimum variance estimator, it operates much as if the pilot carefully plotted his advised position in the yz plane each time he received an advisory, assigned a weight to each datum and computed a weighted least-squares fit to these data. It is doubtful any human pilot can as accurately extract the information content from advisory data. If it becomes advantageous to include other sources of data which the pilot uses to determine glidepath location (such as his visual sighting of the landing strip), these data can also be treated as independent estimates with attached reliability expressed as variances. As an example, if the pilot sees another plane he knows is on the glideslope, this could be treated as an estimate of a linear combination of height and slope with cross-correlation of the errors. The computation in that case becomes slightly more complicated. (The purpose of this example is to show the flexibility of the Kalman filter approach, not to suggest that such varied information sources should be incorporated into the pilot model.)

As presented here, the pilot model only modifies the pilot's conception of where the glideslope is as a result of information received from the controller. He (the pilot) is in that sense entirely dependent on the controller, but in a realistic sort of a way. If the controller gives a string of sound advisories and then gives a bad one, (such as saying "above" when he means "below") the pilot weighs the new advisory against the consistent string of good ones, and only modifies his conceptions in the way which gives best fit to the totality of data. It is, in fact, possible to detect situations like that just described, wherein the pilot would be confronted with new data very inconsistent with the preconceived opinion built upon previous data. The standard deviation of the pilot's estimate of his height above glideslope is $\sqrt{\sigma_{11}}$. If he receives a new estimated position which differs from his own estimate by an amount large compared to the standard deviation of his own current estimate, and also to the standard deviation of the new estimate itself, there is reason to be suspicious of the new data. (The variance of the difference between his current best estimate and any new unbiased and independent estimate is $(\sigma_{11} + V_{ha})$, so the "measure of suspiciousness" is the square of the difference in estimates divided by this variance.)

Any approach to a pilot model which explicitly carries the pilot's conception of where the glideslope is also makes it possible to display that conception to show the trainee the influence of his advisories on the pilot's thinking. A word of caution is advisable in his regard, however. During the initial phases of the approach, the pilot typically has an erroneous impression of where the glideslope is. The straight line in space which is his impression of where he wants to fly deviates from the true glideslope, and in general does not pass through the point of origin used in generating

the display, which has a pseudo-logarithmic scale. Any straight line which does not pass through that point is a highly curved line on the display. As a result of the pseudo-logarithmic display scale, the pilot's early conception of the glideslope would then appear as a curve, deviating wildly from the real glideslope.

Relationship of Estimates to Advisories. Analysis of the display scale, target return size and zone conventions will reveal that each advisory implies that the aircraft is currently within some interval of height relative to the glideslope that is,

$$
\begin{array}{rlrl}
\text{WBGP} & \Rightarrow & & h \le u_{-5}(z) \\
\text{BGP} & \Rightarrow & l_{-3}(z) < & h \le u_{-6}(z) \\
\text{SBGP} & \Rightarrow & l_{-1}(z) < & h \le u_{-4}(z) \\
\text{OG} & \Rightarrow & l_{2}(z) < & h \le u_{-2}(z) \\
\text{SAGP} & \Rightarrow & l_{4}(z) < & h \le u_{1}(z) \\
\text{AGP} & \Rightarrow & l_{6}(z) < & h \le u_{3}(z) \\
\text{WAGP} & \Rightarrow & l_{5}(z) < & h
\end{array}
$$

where the upper and lower limits on h implied by the advisory are functions of the aircraft's z coordinate. (The determination of these highly non-linear functions will fall out of the analysis of display scale in a natural way.) (The indices used in identifying these functions are exactly those used to define zones.)

For those zones with both upper and lower limits it is natural to associate an estimate of height equal to the endpoint of the zone and variance equal to the variance of a random variable uniformly distributed over the zone. Thus in the non-"well" zones, the advisory is treated as an estimate of h with value and variance

$$h_a = (u+l)/2$$
$$V_{ha} = (u-l)^2/12$$

A careful analysis of advisories reveals that each also carries an implication of the pilot's current trend relative to the glideslope, whether or not the message carries a trend statement. Specifically, the absence of a trend statement implies to the pilot that he is moving parallel to the glideslope. The variance of this estimate depends upon both range and his distance from the glideslope. Again, an analysis of the non-linear display scale will reveal that at long ranges the aircraft must move a significant lateral or vertical distance before a trend can be detected, and that the difference between the glideslope and the aircraft's line of descent must also be considerable. At short ranges the content of the trend messages (and the significance of the absence of a trend message) is much greater. Analysis of the display and consultation with controllers and pilots must be used to determine a matrix of values for the trend implication and its variance. The values can be expressed as a difference between the aircraft's current slope of descent and the glideslope slope. The data needed are depicted in table 17.

TABLE 17. MATRIX OF SLOPE AND SLOPE VARIANCE TO BE DEVELOPED

(Some combinations may be meaningless.)

		Trend Information						
		COMING DOWN	GOING FURTHER BELOW	GOING BELOW	NONE	GOING ABOVE	GOING FURTHER ABOVE	COMING UP
Height Zone	WA	$\Delta\xi_a, V\xi_a$						
	A							
	SA							
	OG							
	SB							
	B							
	WB							

An advisory is then interpreted as indicating that the aircraft's path is at slope $\Delta\xi_a$ above the real glideslope, that is, an estimate that the real glideslope is at

$$\xi_a = \Delta\xi_a + \xi \text{ aircraft}$$
$$= \Delta\xi_a - \frac{\dot{y}}{S_H}$$

and the variance of that estimate is $V\xi_a$. However, to estimate the influence of sudden, recent changes in $\dot{y}$ (which an advisory wouldn't necessarily reflect), $\dot{y}$ will be computed as the average rate of descent over the recent past.

Real Pilots. Real pilots deviate from the idealized Kalman model in several ways. More realistic pilot behavior can be obtained by introducing several modifications of the procedure outlined above. Brief descriptions of suggested modifications follow:

1. Random Disregard

   - Absolute

     The pilot will occasionally disregard an advisory for a variety of reasons. With the Kalman model it is adequate to simply ignore the advisory in these cases. The pilot's probability of ignoring an advisory is a natural pilot quality parameter. If the probability of ignoring an advisory is to be greater than about .1 it may be advisable to protect against ignoring two successive advisories in the same (pitch or yaw) channel.

- Relative

  To simulate a pilot's "more or less" responding to the content of an advisory, it is adequate to use an artificially high variance associated with the nominal advisory estimate. This causes the pilot to treat the advisory as an estimate with less than realistic weight. Multipliers of 4-10 should produce noticeable failure to use the full strength of the advisory.

2. Misjudgement of Distance Traveled

The distance traveled in the z direction between advisories (dz) enters into the Kalman calculations. Real pilots have no direct indication of this distance and must infer it, probably intuitively and unconsciously. A random, biased variation from the true distance traveled can be used in the calculations to simulate that aspect of his problem.

3. Misjudgement of the Significance of an Advisory

The values ξ_a and h_a indicate where the advisory implies the real glideslope is. Random variations in these values simulate, for example, a pilot misjudging the significance of "slightly above" relative to "above." A real pilot might think of "slightly above" as indicating that he is about 200 feet above glidepath, irrespective of distance to the landing point, whereas the real significance of this advisory varies a large amount with range. This kind of pilot characteristic can be accomplished by changing (for example) the effective value of the functions $l_4(z)$ and $u_1(z)$ at the beginning of the simulation run. A pilot misinterpretation bias is introduced in that way. Random misinterpretations are also possible, but seem less likely unless highly correlated in time.

4. Injection of Changes Into the Aircraft/Pilot Model

As described in the aircraft/pilot control section, the aircraft/pilot model accepts as input the "setpoint" ξ, which is his opinion of the true glideslope slope. When the Kalman equations are executed to find a new value of ξ, that input value should be changed accordingly. Unlike ξ, h is continually changing due to aircraft motion. The h value to be used in the Kalman equations is therefore its instantaneous value. When the new value is found, the h value in the aircraft/pilot model is instantaneously changed to this new value. In terms of the continuous system model of figure 32, the change can be treated as entry of Δh, where $\Delta h = h^{new} - h^{old}$, at the summing point at the output of the $\dot{h}$ integrator, as illustrated in figure 33.

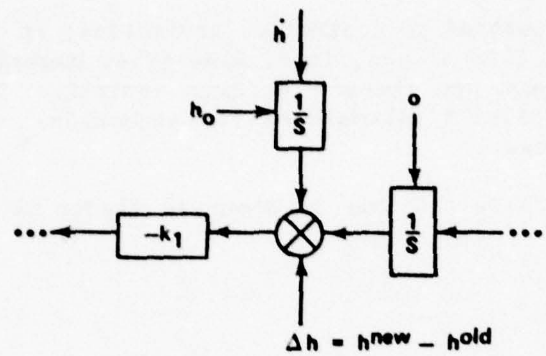


Figure 33. Modification of Continuous System Model of the Aircraft/Pilot Control Simulation to Show Incorporation of Changes in h Computed in Pilot Model

The simulation equations given in the aircraft/pilot control section need not be changed to reflect incorporation of the pilot model, as both ξ and h are simply changed to their new values whenever the pilot model is executed.

Program Descriptions. Starting with a top view description of the APE software, we can identify three major portions: 1) an initializer, 2) a periodic task, and 3) a listening task. The initializer is executed once prior to each simulated landing. Its purpose is to calculate runtime parameter values and to set initial conditions of state variables. It will also set a "new run" flag to indicate the start of a simulated landing, and reset any clocks, counters, or timers.

Some of this initialization activity may be done within the periodic task to avoid cluttering common storage with parameters that are not globally meaningful.

The periodic task is executed every one half second after the handoff to the final controller. It consists of a series of modules which implement dynamic models, i.e., models which simulate continuous systems by updating state variables, usually with small changes at frequent intervals. The wind and aircraft models are included here.

This task will also accumulate seconds since the last advisory in order to create a missed approach if more than five seconds elapse without a message.

The periodic task will also invoke the radar display module (not part of APE) which will draw the target in its latest position.

The listening task responds to controller advisories; it is suspended otherwise. It resets the five second timer, classifies messages in order to set or reset various flags and changes certain variables in the pilot model which represent the pilot's estimate of his situation. It also stores the most recent advisories.

The relationship of these routines is shown in figure 34. Program descriptions follow.

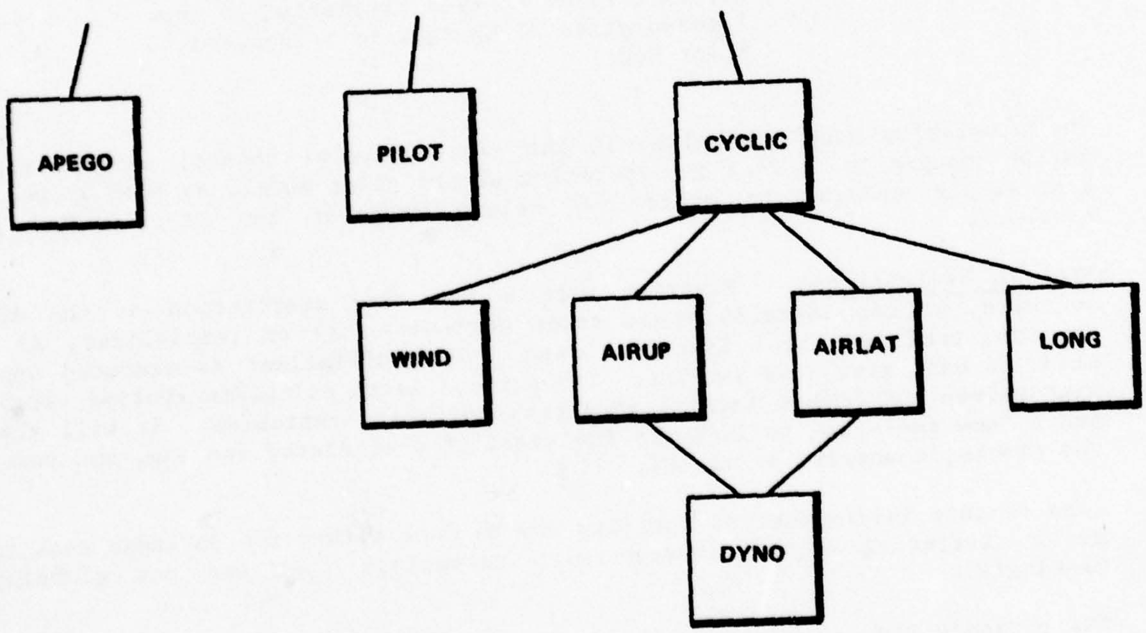


Figure 34. Calling Relationships of APE Components

APEGO

Description: Initializer for APE.

Entry point: APEGO

Classification: Subroutine

Period: Once per simulated landing

Language: F

Activated/called by: Phase Executive

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: ACFIX, ENVIRON, PLT

Files created/changed: None

Files referenced: None

PILOT

Description: Listening task which processes controller advisories.

Entry point: Pilot

Classification: Task

Period: Non-Periodic

Language: F

Activated/called by: Speech Understanding

Cancelled by: Self

Activates/calls: None

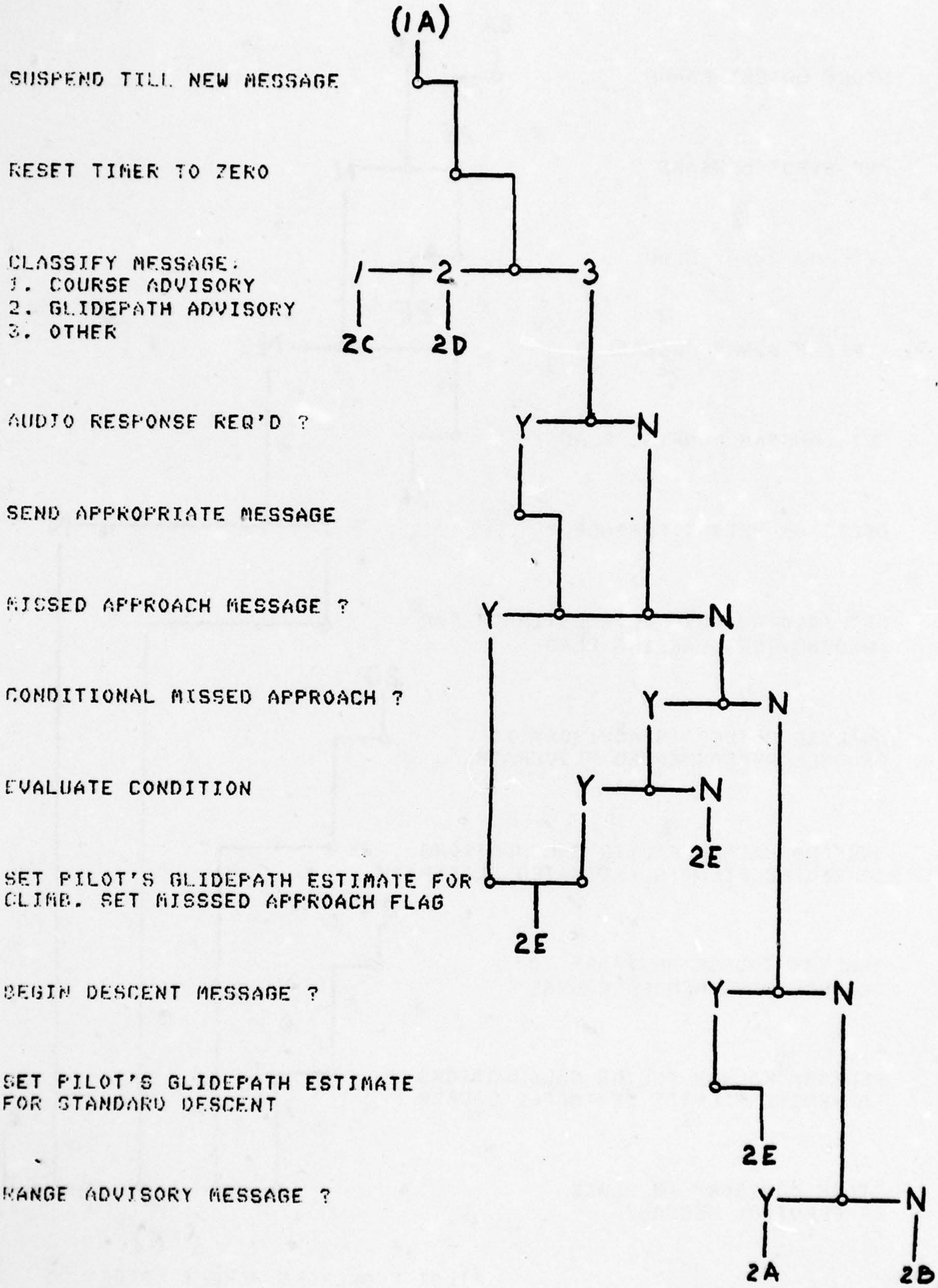
Cancels: None

Input arguments: Speech identification words

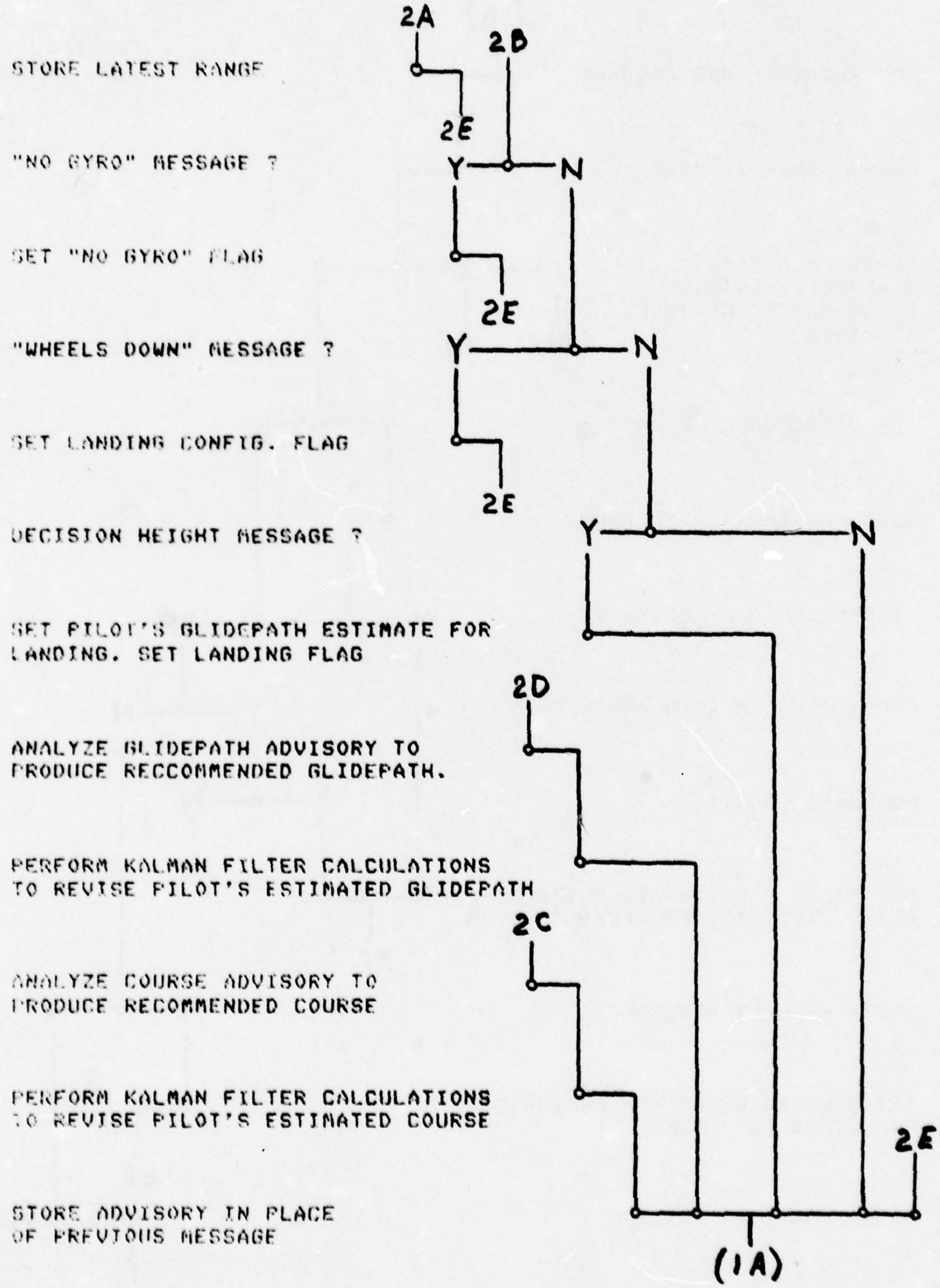
Common variables: Various condition flags and pilot model variables in blocks PLT and ACFIX

Files created/changed: None

Files referenced: None



PILOT FLOWCHART (SHEET 1 OF 2)



PILOT FLOWCHART (SHEET 2 OF 2)

CYCLIC

Description: Periodic task to call dynamic models and radar display.

Entry point: CYCLIC

Classification: Task

Period: 1/2 second

Language: F

Activated/called by: Phase Executive

Cancelled by: Self

Activates/calls: AIRUP, AIRLAT, WIND, RADAR, LONG

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

WIND

Description: Wind simulation (dynamic, stochastic).

Entry point: Wind

Classification: Subroutine

Period: 1/2 second

Language: F

Activated/called by: Periodic Task

Cancelled by: N/A

Activates/calls: None

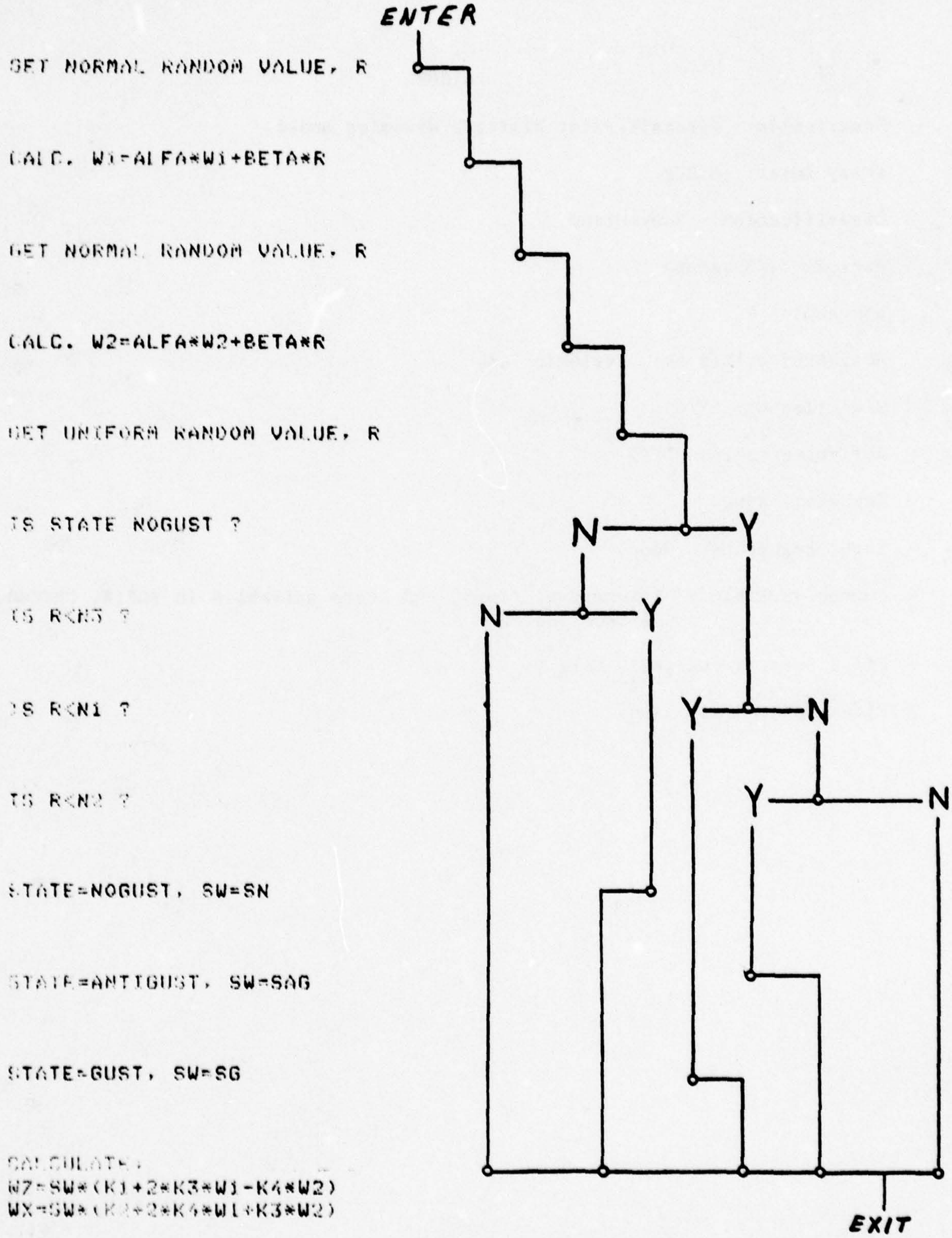
Cancels: None

Input arguments: None

Common variables: Wind components and random variables in ENVIRON block

Files created/changed: None

Files referenced: None



WIND FLOWCHART (SHEET 1 OF 1)

AIRUP

Description: Aircraft/Pilot altitude dynamics model.

Entry point: AIRUP

Classification: Subroutine

Period: 1/2 second

Language: F

Activated/called by: Periodic Task

Cancelled by: N/A

Activates/calls: DYNO

Cancels: None

Input arguments: None

Common variables: Parameters, flags, and state variables in ACFIX, EMERGE, ENVIRON and PLT

Files created/changed: None

Files referenced: None

IS "NEW RUN" FLAG SET ?

CALCULATE RUNTIME PARAMETERS,
RESET "NEW RUN" FLAG

IS "TOUCH DOWN" FLAG SET ?

CALCULATE INTERMEDIATE EXPRESSIONS

CALCULATE RATES OF CHANGE
OF STATE VARIABLES

UPDATE STATE VARIABLES

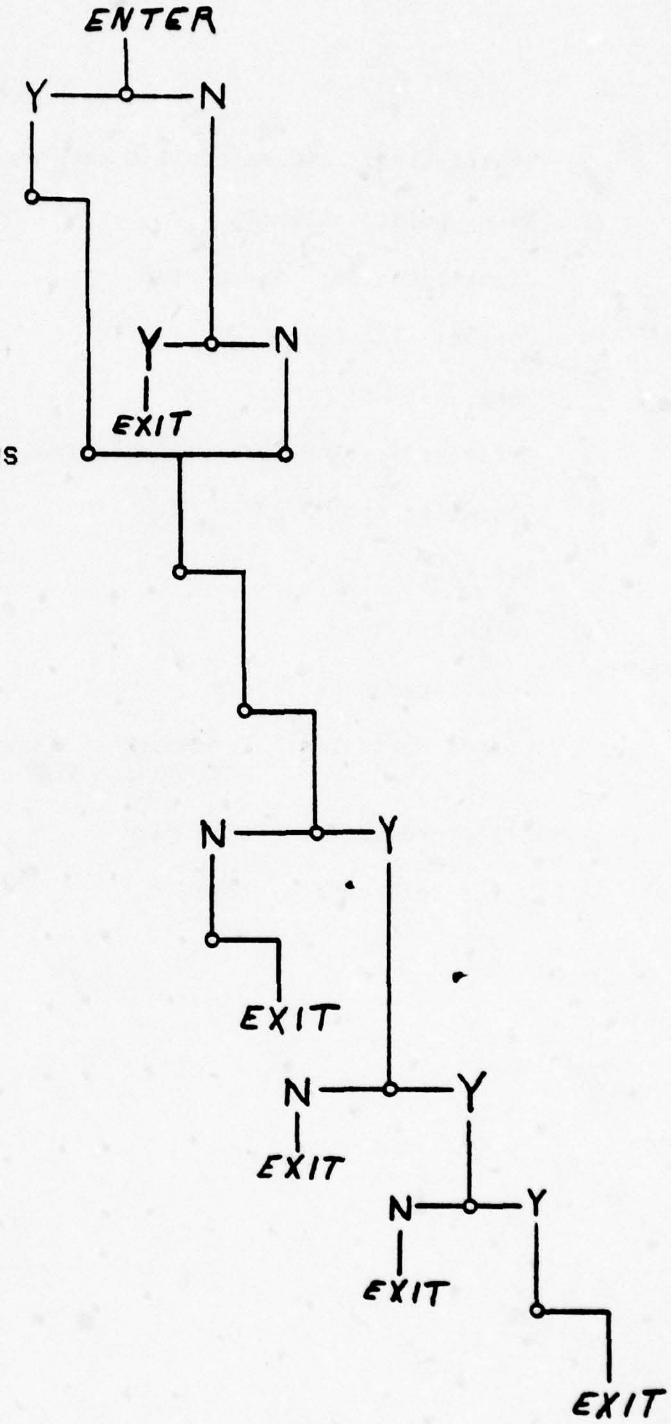
ALTITUDE > ZERO ?

ALT.=0, RATE OF DESCENT=0,
SET "TOUCH DOWN" FLAG

BELOW VISIBILITY CEILING ?

EMERGENCY PULLUP REQ'D ?

PERFORM ALTERNATE STATE UPDATE
SET MISSED APPROACH FLAG
SET PILOT'S ESTIMATED GLIDEPATH
FOR MISSED APPROACH



AIRUP FLOWCHART (SHEET 1 OF 1)

AIRLAT

Description: Aircraft/pilot lateral dynamic model.

Entry point: AIRLAT

Classification: Subroutine

Period: 1/2 second

Language: F

Activated/called by: Periodic Task, CYCLIC

Cancelled by: N/A

Activates/calls: DYNO

Cancels: None

Input arguments: None

Common variables: Parameters, flags and state variables in ACFIX, EMERGE, ENVIRON and PLT

Files created/changed: None

Files referenced: None

LONG

Description: Aircraft longitudinal dynamics model.

Entry point: LONG

Classification: Subroutine

Period: 1/2 second

Language: F

Activated/called by: Periodic Task

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: Parameters and state variables in ACFIX and ENVIRON

Files created/changed: None

Files referenced: None

IS "NEW RUN" FLAG SET ?

CALCULATE APPROACH GND SPD,
RESET "NEW RUN" FLAG

SET ACCELERATION TO ZERO

IS GROUNDSPEED > APPROACH GND SPD ?

IS RANGE GREATER THAN
LANDING CONFIGURATION RANGE ?

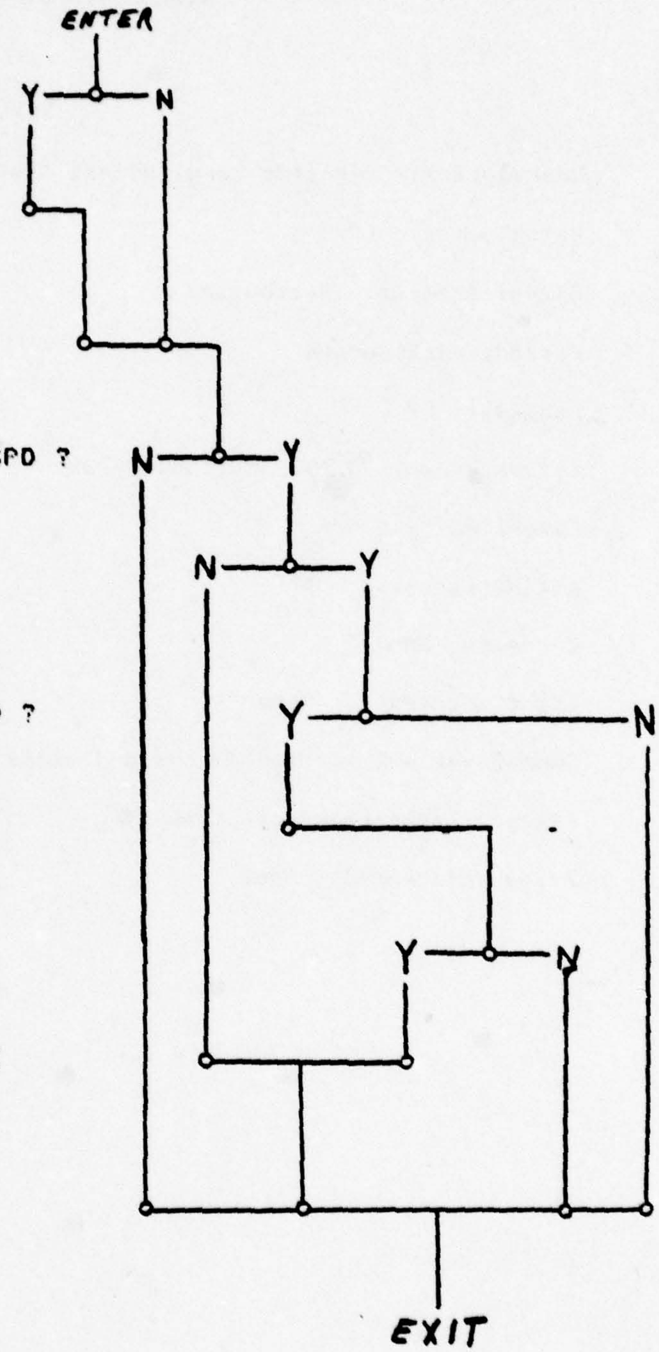
HAS "WHEELS DOWN" BEEN RECEIVED ?

GET UNIFORM RANDOM VALUE.

DECIDE IF PILOT WILL RESPOND

SET ACCELERATION TO APPROPRIATE
VALUE FOR A/C TYPE

UPDATE STATE VARIABLES
GROUNDSPEED AND RANGE



LONG FLOWCHART (SHEET 1 OF 1)

DYNO

Description: Generalized third order aircraft dynamics model.

Entry point: DYNO

Classification: Subroutine

Period: Twice every 1/2 second

Language: F

Activated/called by: AIRUP and AIRLAT

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: Parameters, flags and state variables in ACFIX, EMERGE, ENVIRON and PLT

Files created/changed: None

Files referenced: None

Radar

The radar simulation, residing on CPU1, is responsible for several areas. The subroutine receives arguments from two sources, APE and SERVO. APE calls RADAR approximately every half second, providing information on target position in miles, in terms of course, altitude and range. This information is converted to screen coordinates by the routine LOOKUP. LOOKUP uses a table which translates linear distances to relative logarithmic distances. Each time a new target position is received in the difference between the log of the old position and the log of the new position is calculated. Since the distance a target will travel is quite small, the table required for screen coordinate conversion will use less space and permit faster execution.

Once the new target position is calculated, TRGSZ is called. Target returns shrink as they move down the glidepath. This routine uses another table to determine how large the target should be, according to how far away it is. Range is sent over as the first argument. The coordinates returned are the top and bottom of the target. Since azimuth and elevation targets are always the same size, TRGSZ need only be called once. Once this information has been obtained, it is sent to the aircraft update routine, PICUP, via the interprocessor bus (IPB).

SERVO, which resides on CPU2, communicates with RADAR across the IPB. It sends information on the servo position. RADAR uses this information to determine the visibility of the target return. If the target is not in the servo beam, it will not be visible. RADAR also uses the servo position with antenna alignment information provided by phase 1 in common to determine the visibility of the centerline and touchdown reflectors. Both the servo information and the target return screen coordinates are stored on the disk and in common for later use by the REPLAY routine.

TRGSZ

Description: Determines target size in screen coordinates from a table using the range of the target for further information.

Entry point: TRGSZ

Classification: Subroutine

Period: None

Language: F

Activated/called by: RADAR

Cancelled by: N/A

Activates/calls: None

Cancels: None

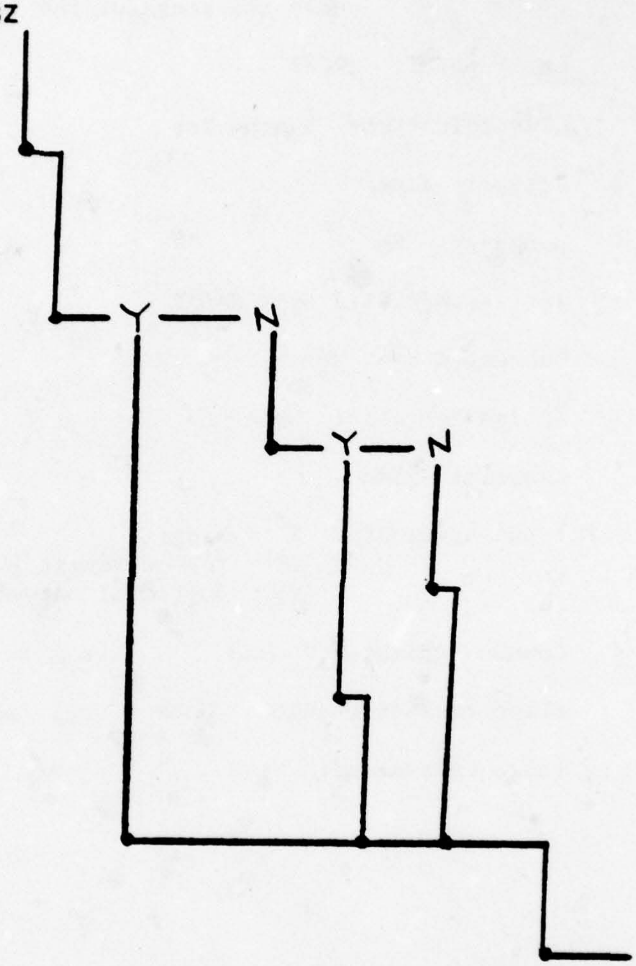
Input arguments: X — Range
YT — Top of target (Y1 or Y3)
YB — Bottom of target (Y2 or Y4)

Common variables: None

Files created/changed: None

Files referenced: None

TRGSZ



LOOK UP YT IN TABLE FOR TARGET SIZE. DEPENDENT ON RANGE, X

DO COORDINATES OF TARGET FIT WITHIN SERVO BEAM?

IS TARGET AT ALL WITHIN THE BEAM?

YT = 32000

TRUNCATE TARGET

STORE TARGET COORDINATES IN YT, YB

RETURN

TRGSZ FLOWCHART (SHEET 1 OF 1)

LOOKUP

Description: Converts linear miles to screen coordinates in a logarithmic display.

Entry point: LOOKUP

Classification: Subroutine

Period: None

Language: F

Activated/called by: RADAR

Cancelled by: N/A

Activates/calls: None

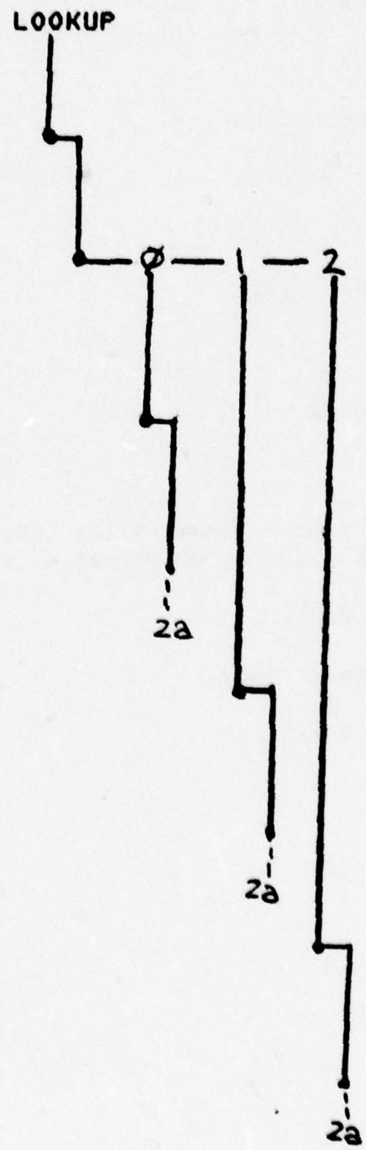
Cancels: None

Input arguments: MILES — linear miles (input)
X — screen coordinates (output)

Common variables: RDMIL

Files created/changed: None

Files referenced: None



FIND LOG OF MILES

BRANCH ON MSG 0-X
1-Y1
2-Y3

SUBTRACT LOG (MILES) FROM RALOG

STORE LOG (MILES) IN RALOG

SUBTRACT LOG (MILES) FROM AZLOG

STORE LOG (MILES) IN AZLOG

SUBTRACT LOG (MILES) FROM ELOG

STORE LOG (MILES) IN ELOG

LOOKUP FLOWCHART (SHEET 1 OF 2)

2a

LOOK UP DIFFERENCE OF LOGS IN TABLE FOR SCREEN COORDINATES

STORE IN XY1Y3

RETURN

LOOKUP FLOWCHART (SHEET 2 OF 2)

SERVO

Description: Causes misalignment and resets alignment. Causes movement of antenna. Returns servo position update.

Entry point: SERVO

Classification: Task

Period: .5 second

Language: F

Activated/called by: IPBIN2

Cancelled by: N/A

Activates/calls: IPBOUT2

Cancels: None

Input arguments: MSG — Code to determine what is to be done
X — zone to which to move reflector
Y — position of servo in x- or y-plane

Common variables: SVLOY, SVLOX, SVHIY, SVHIX, SVSETX, SVSETY, SVZN(4), SVSTDA(2), SVSTDE(2), SVSCL(2)

Files created/changed: None

Files referenced: None

PROCEDURE

BRANCH ON MSG #

DOES Y = 32000?

STORE Y IN SVSETX

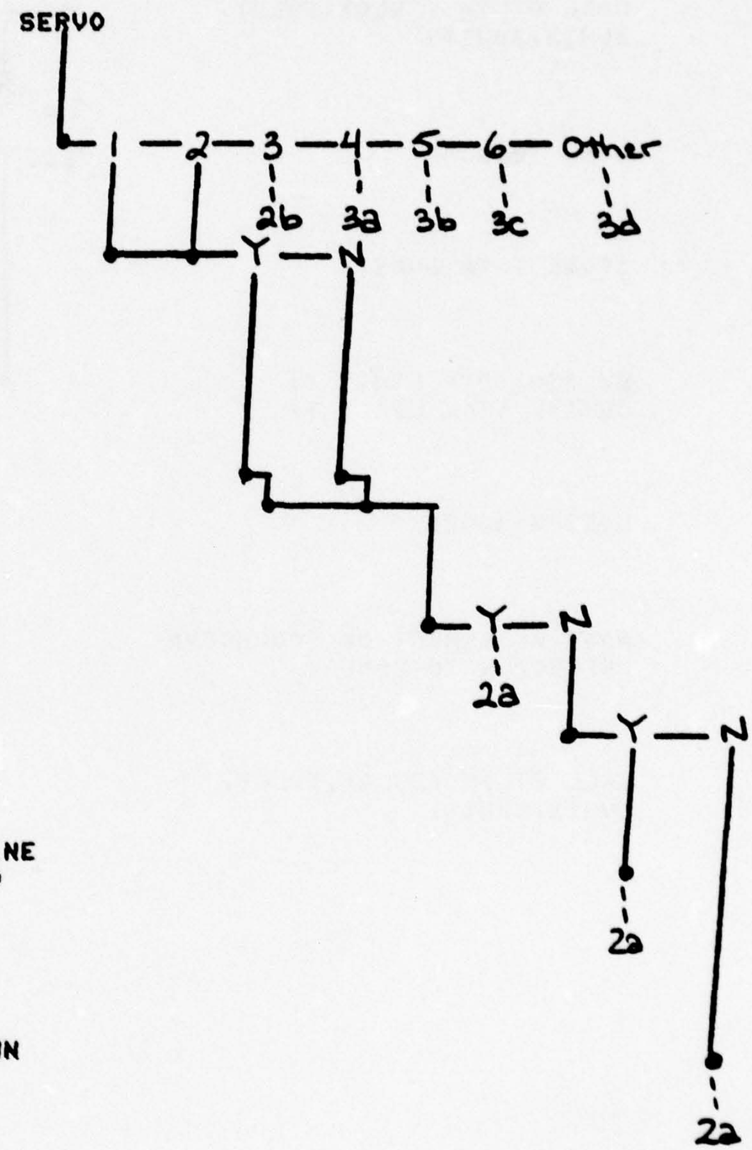
SVLOX= (MIN LIMIT,X)
SVHIX= (MAX LIMIT,X)

DOES X=32000?

DOES MSG=1?

MOVE ALIGNMENT OF CENTERLINE REFLECTOR TO ZONE X. (ADD DISPLACEMENT SVZN # (Y) TO SVSTD)

MOVE ALIGNMENT OF TOUCHDOWN REFLECTOR TO ZONE X



SERVO FLOWCHART (SHEET 1 OF 3)

CALL GTLIM (SVLOX,SVLOY,
SVHIX,SVHIY)

DOES Y=32000?

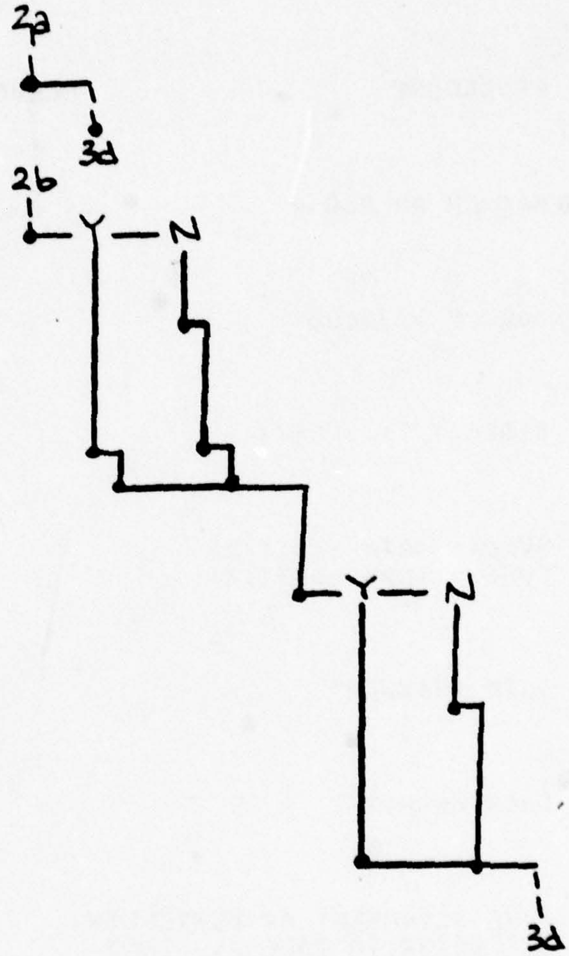
STORE Y IN SVSETY

SVLOY= (MIN LIMIT Y)
SVHIY= (MAX LIMIT Y)

DOES X=32000?

MOVE ALIGNMENT OF TOUCHDOWN
REFLECTOR TO ZONE X

CALL GTLIM (SVLOX,SVLOY,
SVHIX,SVHIY)



SERVO FLOWCHART (SHEET 2 OF 3)

3a

SVLOX= CENTERPOINT
SVHIX= CENTERPOINT

CALL GTLIM (SVLOX,SVLOY,
SVHIX,SVHIY)

3e

3b

SVLOY= CENTERPOINT
SVHIY= CENTERPOINT

CALL GTLIM (SVLOX,SVLOY,
SVHIX,SVHIY)

3e

3c

MOVE HASHMARKS WITHOUT ACTI-
VATING SERVO TO THE POSITION
AT WHICH THEY WOULD BE
IF SERVO WAS AT POSITION
(YA,YE)

3e

3d

TYPE "ERROR-INVALID MESSAGE
#-SERVO"

CALL KILL

3e

SERVO FLOWCHART (SHEET 3 OF 3)

RADAR

Description: Determines visibility and location of target return in screen coordinates. Determines visibility of centerline and touchdown reflectors. Updates common block (location of target and servo position).

Entry point: RADAR

Classification: Subroutine

Period: None

Language: F

Activated/called by: APE, SERVO

Cancelled by: N/A

Activates/calls: PICUP, IMAGES, LOOKUP, TRGSZ

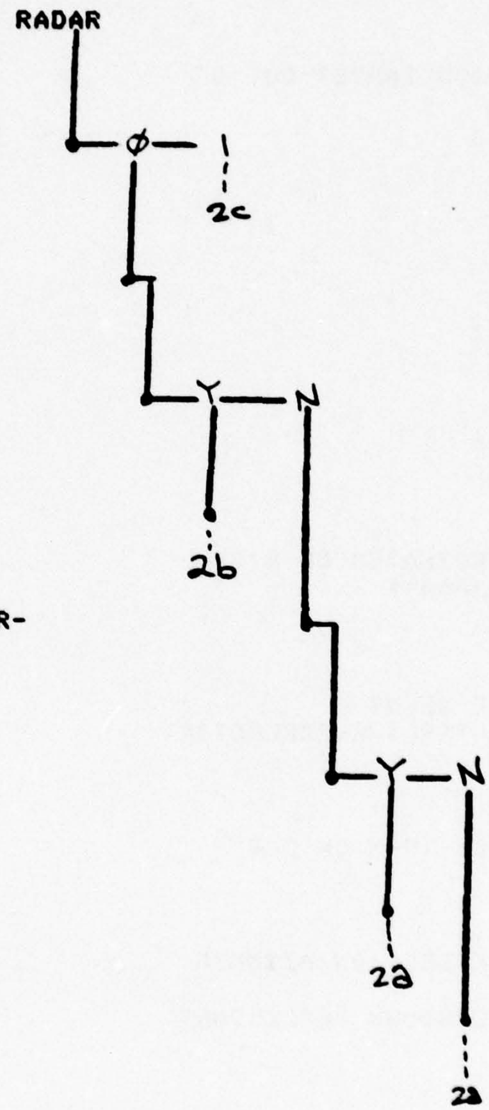
Cancels: None

Input arguments: MSG — indicates who called Radar
X — range in miles
Y_A — screen coordinates of servo-azimuth
Y_E — screen coordinates of servo-elevation

Common variables: RDSVAXZ, RDSVEL, RDCLR, RDTDR, RDRNG, RDALT, RDCRS, RDRAN

Files created/changed: None

Files referenced: None



BRANCH ON MSG # 0-FROM APE
1-FROM SERVO

CALL LOOKUP FOR RANGE

ARE TARGETS OUT OF RANGE?

X=32000

CALL LOOKUP FOR AZIMUTH TARGET

IS AZIMUTH TARGET OUT OF RANGE?

Y1=32000

CALL TRGSZ

RADAR FLOWCHART (SHEET 1 OF 3)

CALL LOOKUP FOR ELEVATION TARGET

IS ELEVATION TARGET OUT OF RANGE?

Y3=32000

CALL TRGSZ

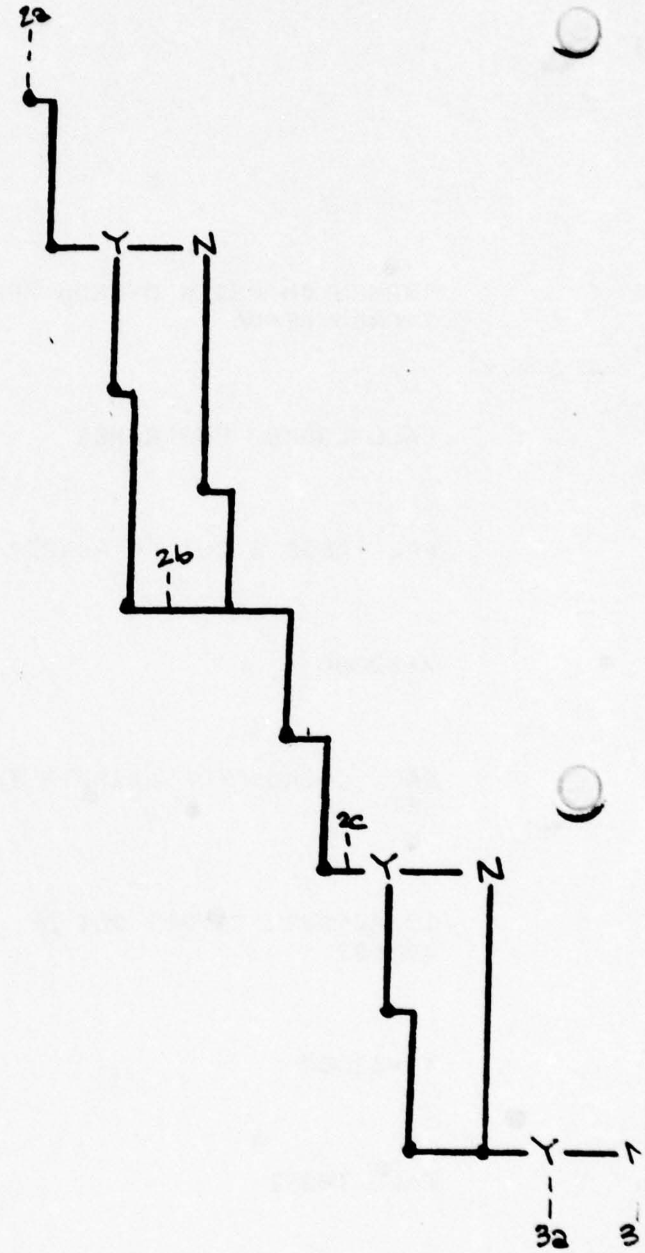
TASK PICUP

STORE COORDINATES ON DISK AND IN COMMON

CAN CLR BE SEEN?
(CLR - CENTERLINE REFLECTOR)

TASK IMAGES-TURN ON CLR

CAN TDR BE SEEN ON AZIMUTH DISPLAY?
(TDR - TOUCHDOWN REFLECTOR)



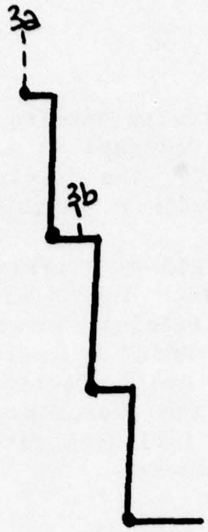
RADAR FLOWCHART (SHEET 2 OF 3)

TASK IMAGES-TURN ON TDR ON
AZIMUTH DISPLAY

CAN TDR BE SEEN ON ELEVATION
DISPLAY?

TASK IMAGES, TURN ON TDR ON
ELEVATION DISPLAY

RETURN



RADAR FLOWCHART (SHEET 3 OF 3)

Display

The display software handles the simulation of the PAR display as well as other pictures designed to enhance the training process. The software is resident on CPU2. The display is manipulated by a series of message codes designed to provide a flexible environment for the training system.

Display is managed by a task called IMAGES. This task turns relevant pictures on and off. IMAGES also calls a routine, RESET, to set targets and trails back to original intensities and prepare these pictures for a fresh run. Each time RESET is called, the PAR display is moved slightly. This is a precaution to avoid drawing a picture too many times in the same place and thus weakening the phosphor. The program may be called at other times as required. The following pictures are implemented at various times during the training process.

1. Azimuth display
2. Azimuth hashmarks
3. Elevation display
4. Elevation hashmarks
5. Glidepath zone positions
6. Course zone positions
7. Sweep
8. Azimuth target
9. Azimuth trail
10. Elevation target
11. Elevation trail
12. ICS panel
13. Keyboard

Other routines are also necessary for the display software. PICUP is designed to update the location of the target return and position the trail. This routine is capable of displaying the target with a 1 1/2 inch trail normally seen on a PAR scope, or a long trail, which is used during the training process to demonstrate the target path to the student controller. Another routine, called by PICUP, FADOFF, fades trails realistically, or fades to the long trail to show the target path. SERVO, the joystick routine, handles both the alignment and the location of each antenna. The position of the servo determines whether a target will be visible. During runs, the student will be able to manipulate the servo to keep the target in sight.

At other times in the training process, such as during replay runs, either antenna may be manipulated without student control. The servo routine will also return servo position updates to the radar simulator which is resident on CPU1.

RESET

Description: Resets display list to initial values.

Entry point: RESET

Classification: Subroutine

Period: None

Language: F

Activated/called by: IMAGES, SERVO

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

RESET

SET TARGETS TO INITIAL VALUES

SET TRAILS TO INITIAL VALUES

MOVE DISPLAY TO AVOID BURNING PHOSPHOR

ALIGN CLR & TDR TO NEW DISPLAY

ALIGN SERVO LIMITS TO NEW DISPLAY

RETURN

RESET FLOWCHART (SHEET 1 OF 1)

IMAGES

Description: Handles overlaps, if any. Turns pictures on and off. Turns display processor on and off. Calls for initialization of display list when first activated.

Entry point: IMAGES

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN2

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: I — message type
J — picture, or 0 to refer to display processor

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE

DOES J=0? (IF J≠0, THEN J= PICTURE NUMBER)

DOES I=3?

BLINK PICTURE X

CALL PONOF (J,I)

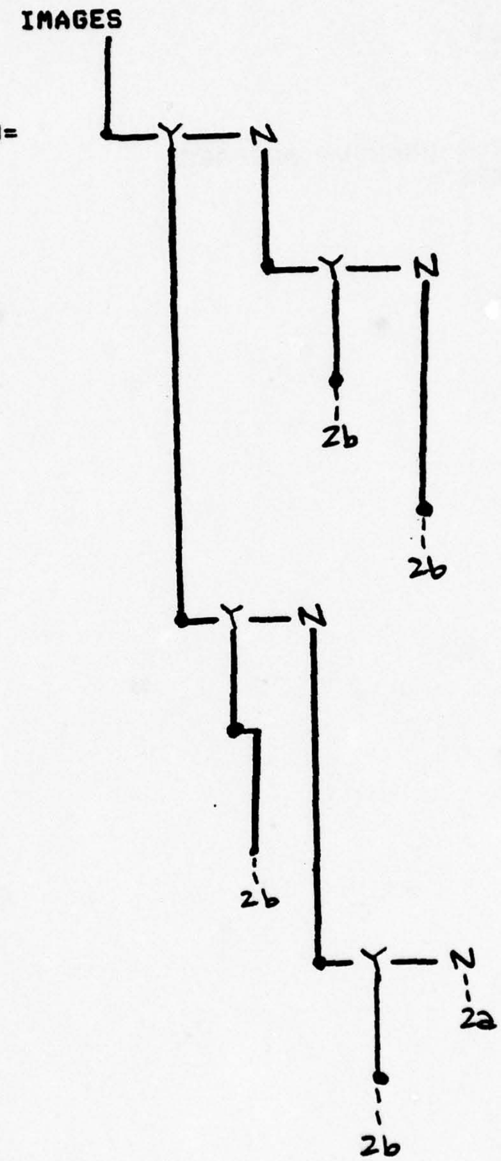
DOES I=2?

TURN OFF ALL PICTURES

CALL RESET

DOES I=5?

CALL DSTRT



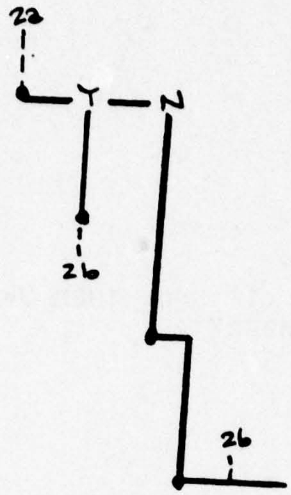
IMAGES FLOWCHART (SHEET 1 OF 2)

DOES I=6?

CALL DHALT

TYPE "ERROR-UNKNOWN MESSAGE
CODE-IMAGES"

CALL KILL



IMAGES FLOWCHART (SHEET 2 OF 2)

FADOFF

Description: Fades trails.

Entry point: FADOFF

Classification: Task

Period: None

Language: F

Activated/called by: PICUP

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: X — fade to long trails = 1; fade out = 0

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE FADOFF

IS FIRST INTENSITY NEGATIVE?

MOVE INTENSITY Y-1 TO Y

IS THAT THE LAST POSITIVE INTENSITY?

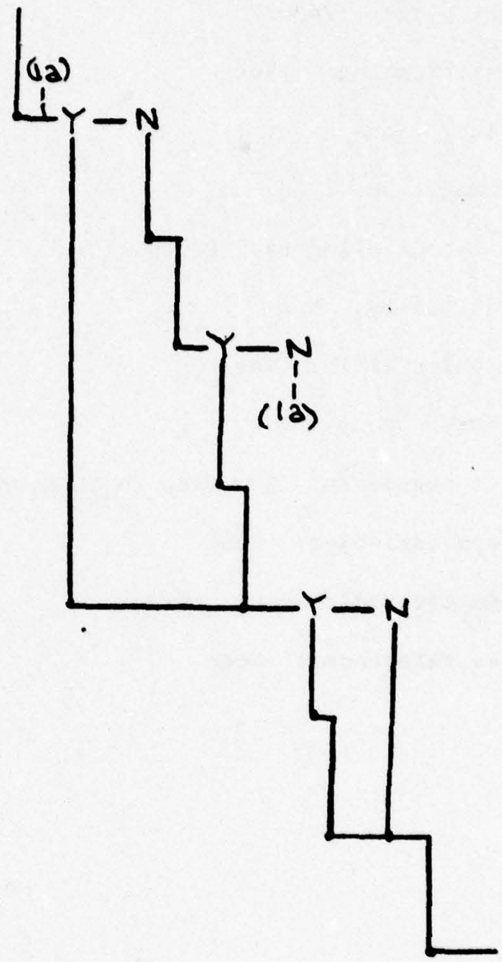
NEGATE IT

DOES X=1?

DISPLAY LONG TRAIL

RESET SHORT TRAIL

CALL KILL



FADOFF FLOWCHART (SHEET 1 OF 1)

PICUP

Description: Picture update — moves targets/trails. Calls for fade trails. Images is presumably invoked previous to this task so that appropriate pictures are on.

Entry point: PICUP

Classification: Task

Period: .5 sec

Language: F

Activated/called by: IPBIN2

Cancelled by: N/A

Activates/calls: Fadoff

Cancels: None

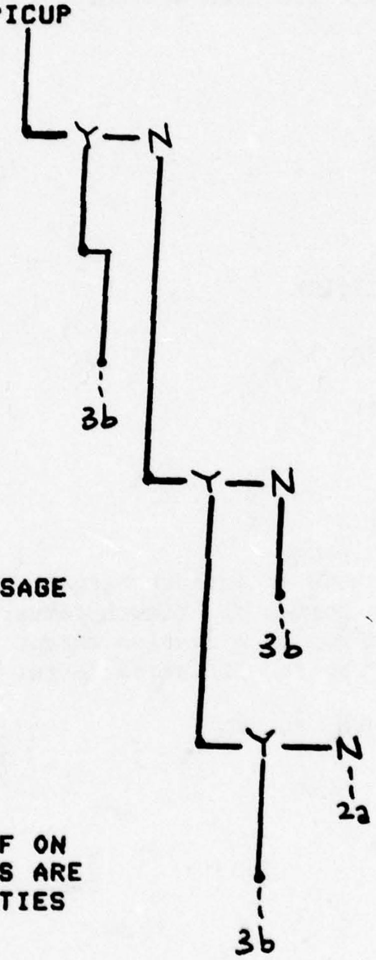
Input arguments: MSG —
- X — range
- Y1 — top of azimuth target
- Y2 — bottom of azimuth target
- Y3 — top of elevation target
- Y4 — bottom elevation target

Common variables: PCAMVD, PCEMVD

Files created/changed: None

Files referenced: None

PROCEDURE PICUP



DOES MSG=4?

CALL FADOFF (X)

CALL RESET

DOES MSG=1 OR 2?

TYPE "ERROR-INVALID MESSAGE CODE-PICUP"

DOES X=32000?

TURN ALL INTENSITIES OFF ON TARGETS. IF INTENSITIES ARE POSITIVE, LOWER INTENSITIES ON TAILS.

PICUP FLOWCHART (SHEET 1 OF 3)

DOES Y1=32000?

TURN OFF ALL INTENSITIES ON AZIMUTH TARGET. LOWER INTENSITIES OF AZIMUTH TAILS.

ADD CHANGE OF X AND CHANGE OF Y TO EACH AZIMUTH TARGET

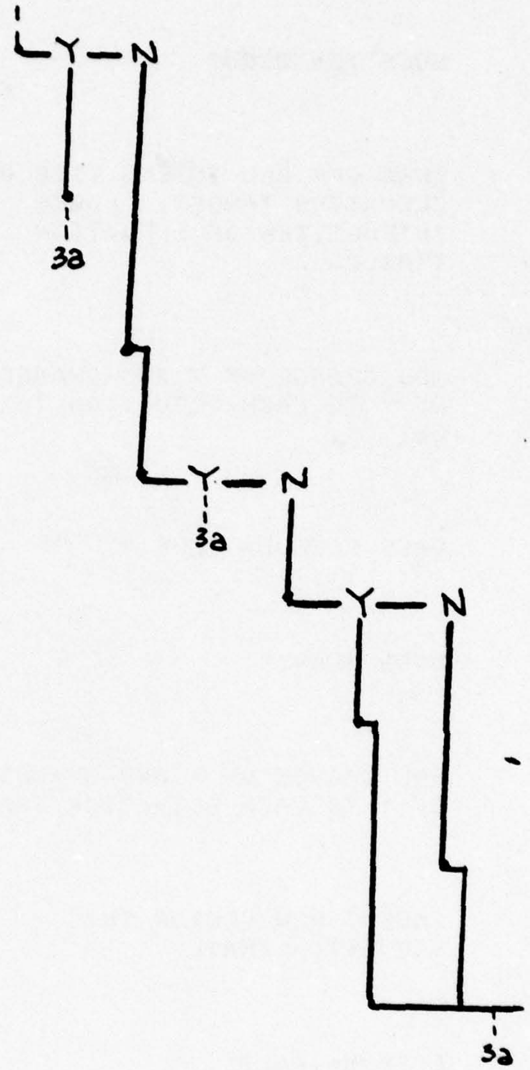
DOES PCAMVD=.TRUE.?

DOES MSG=1?

ADD CHANGE OF X AND CHANGE OF Y TO EACH AZIMUTH TRAIL

INSERT NEW VECTOR IN AZIMUTH TRAIL

PCAMVD=.FALSE.



PICUP FLOWCHART (SHEET 2 OF 3)

DOES Y3=32000?

TURN OFF ALL INTENSITIES ON ELEVATION TARGET. LOWER INTENSITIES OF ELEVATION TRAILS

ADD CHANGE OF X AND CHANGE OF Y TO EACH ELEVATION TARGET

DOES PCEMVD=.TRUE.?

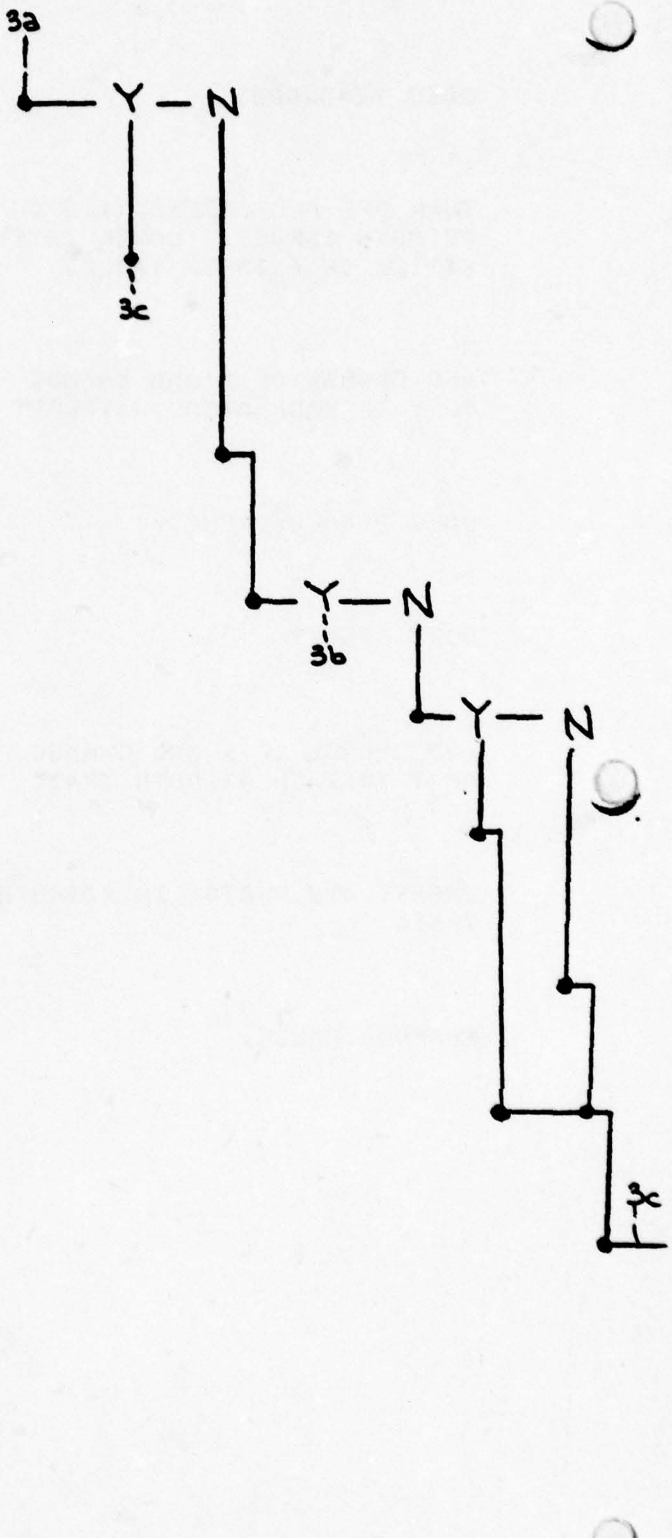
DOES MSG=1?

ADD CHANGE OF X AND CHANGE OF Y TO EACH ELEVATION TRAIL

INSERT NEW VECTOR IN ELEVATION TRAIL

PCEMVD=.FALSE.

CALL KILL



PICUP FLOWCHART (SHEET 3 OF 3)

CREATE

Description: Creates original display list with all pictures turned off.

Entry point: CREATE

Classification: Subroutine

Period: None

Language: F

Activated/called by: CPU2 Initialization

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE CREATE

INITIALIZE DISPLAY LIST

CREATE AZIMUTH DISPLAY (1)

CREATE AZIMUTH HASHMARKS (2)

CREATE ELEVATION DISPLAY (3)

CREATE ELEVATION HASHMARKS (4)

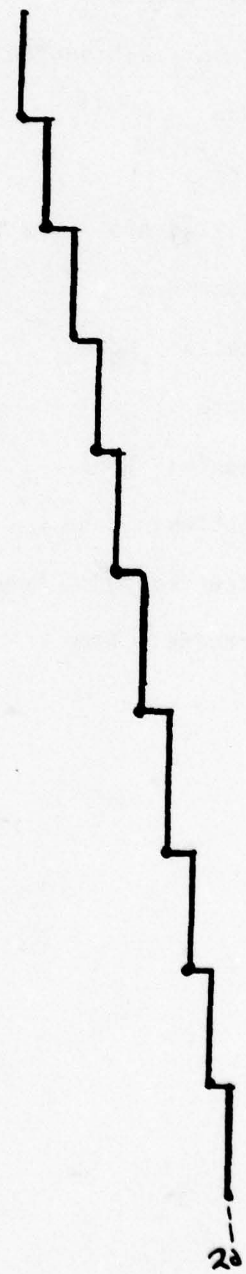
CREATE GLIDEPATH ADVISORIES (5)

CREATE COURSE ADVISORIES (6)

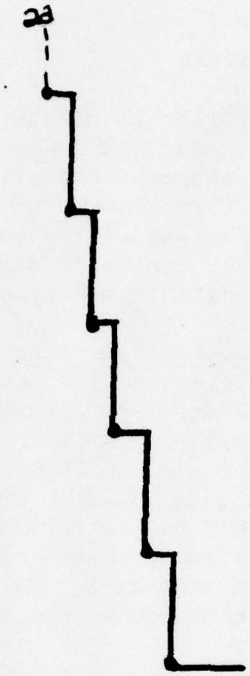
CREATE SWEEP (7)

CREATE AZIMUTH TARGET (8)

CREATE AZIMUTH TRAIL (9)



CREATE FLOWCHART (SHEET 1 OF 2)



CREATE ELEVATION TARGET (10)

CREATE ELEVATION TRAIL (11)

CREATE ICS PANEL (12)

CREATE KEYBOARD (13)

TURN OFF ALL PICTURES

RETURN

CREATE FLOWCHART (SHEET 2 OF 2)

Model Controllers

A final controller is simulated and is used to provide demonstrations. The model final controller is a sophisticated simulation of an ideal PAR controller. It chooses advisories using the same rules the trainee is taught, speaks them, then conveys them to the pilot after the speech output is complete. Simulations of the pattern controller and tower controller are also provided. They are quite simple, but are capable of different behaviors as specified by data in the problem specification files.

Final Model Controller. The final model controller is a series of routines designed to determine the best possible message for a given situation. This model controller is activated during phase 1, and when demonstrations are being given for the student controller. The routines are controlled by a master routine called EXPERT, which activates other routines to establish the priority list, handle timing, perform calculations, and build the message to be sent to the VOTRAX via GLIB, the message output controller. The relationships among these routines are shown in figure 35. Once the VOTRAX has output the message, APE is notified so that the aircraft simulation performs in the appropriate manner.

Once EXPERT has been activated, it activates a routine called RANK. RANK determines the messages appropriate for a given situation and ranks them in order of priority. Under most conditions the message priorities are as follows:

1. Mandatory missed approaches
2. Low altitude alert
3. Optional missed approaches
4. At decision height
5. Vectors
6. Trend and glidepath calls
7. Range calls
8. Wind and clearance calls
9. Course calls
10. Handoffs

This ranking is dependent in part on message groups which are stored in common. The basic groups are shown below.

1. Glidepath position
2. Glidepath trend

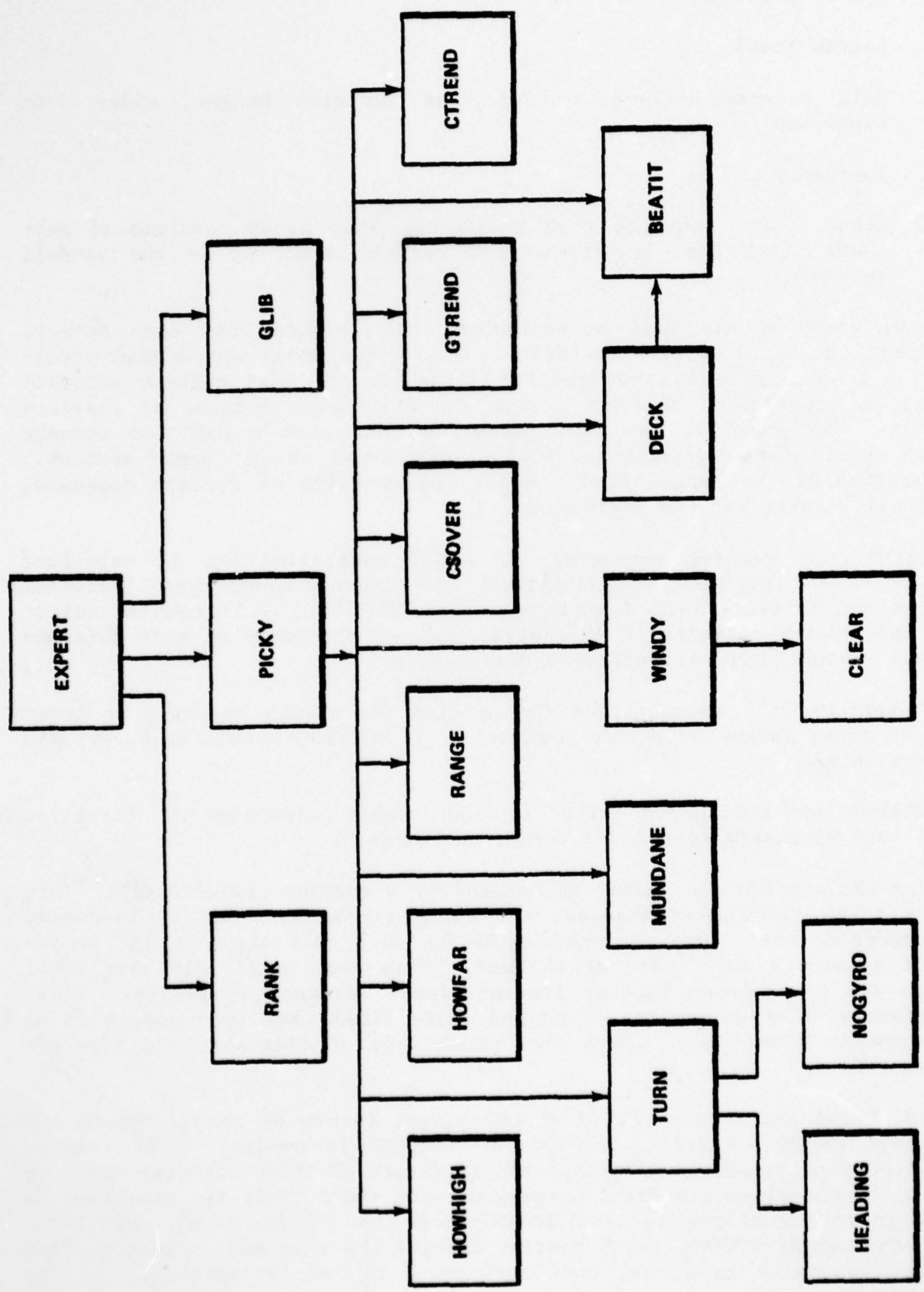


Figure 35. Final Controller Model Design

3. Course position

4. Course trend

5. Mile related advisories (e.g., at decision height, miles from touchdown)

6. Emergency

7. Other (e.g., sequence type messages. This group consists of messages which must be given at a certain time, as in the handoff sequence).

Once this ranking has been accomplished, the routines for each message groups are called in order of priority. If a given group with a high priority is not applicable to the specific situation the next highest priority routine is activated. Certain groups are eliminated because of previous messages. For example, two trend messages must have a position message between them. Some messages are spoken only once, as in "begin descent." The location of the target also changes the priority of certain messages, specifically range related advisories.

In addition to ranking messages, the model controller has to calculate headings, allow for weather fluctuations and properly handle gyro failures. Headings are calculated in a routine called HEADING. This routine determines the present position of the target and, on the basis of that information and weather data, calculates a new heading.

NOGYRO performs the calculations and handles the timing required to start and stop turns during a no-gyro approach. If NOGYRO is activated, HEADING is deactivated.

Both HEADING and NOGYRO are called by TURN, which calculates the direction of the turn necessary to put the target on course.

Building the message for output is managed by a routine called PICKY. This routine takes the chosen message, and calls routines to attach headings, wind speeds or call signs as needed. CSOVER puts call signs on the beginning of a message and "over" at the end. This routine is only used until the "do not acknowledge further transmissions" advisory is spoken. WINDY determines wind speed and direction and calls CLEAR when the message to be sent involves clearance. CLEAR adds which type of clearance the aircraft has received.

HOWHIGH checks the zone position of the target return on the glidepath and determines whether a glidepath position message is needed. This routine must allow for the restriction that two glidepath position messages never be given without a glidepath trend message between them. This is guaranteed by a flag in common shared between HOWHIGH and the routine which calculates glidepath trends, GTREND. Each routine changes the flag and no routine will set the flag twice in a row, thus conforming to the restriction. In the situation where a target is well above glidepath, this restriction is not

valid. GTREND determines what a target's trend is by checking its coordinates. If the target's upper or lower third changes zones, a trend msssage is transmitted subject to the previously mentioned restrictions. Position message routines are called before trend message routines, therefore the trend routines do not perform the position/trend check.

HOWFAR, the course position routine, and CTREND, the course trend routine, behave very similarly to HOWHIGH and GTREND. The same kinds of restrictions apply, and are handled in much the same way. The ratio of course to glidepath messages should be about 1 to 3. These messages are discontinued at landing threshold. DECK, the decision height routine determines if the aircraft is too far off course or glidepath, and may call BEATIT, to conduct a waveoff. BEATIT handles all waveoffs, mandatory or optional.

All information needed by the model controller is stored in common, filled by phase 1, APE and other routines. The model controller in turn fills common with the index of the message sent, so that the target will move in the appropriate manner. Each time a message is built, it is sent to GLIB in an array, each element of which is a phrase index number. The entire message is thus in VSCON format. GLIB will output this array up to the first zero, which terminates the message output. When the VOTRAX is finished with a message, a done flag is set. The model controller then notifies APE, and begins the process again. The model controller must output a message at least once every five seconds with one second between them. Using this method will satisfy this requirement.

Following the descriptions of these routines are the descriptions of the routines which fill the queue of advisories for range related advisories and sequence messages.

EXPERT

Description: Final controller task which activates all final controller modules which select the best advisory.

Entry point: EXPERT

Classification: Task

Period: None

Language: F

Activated/called by: PHAZ1, DEMO

Cancelled by: DEMO, PHAZ1

Activates/calls: RANK, PICKY, GLIB

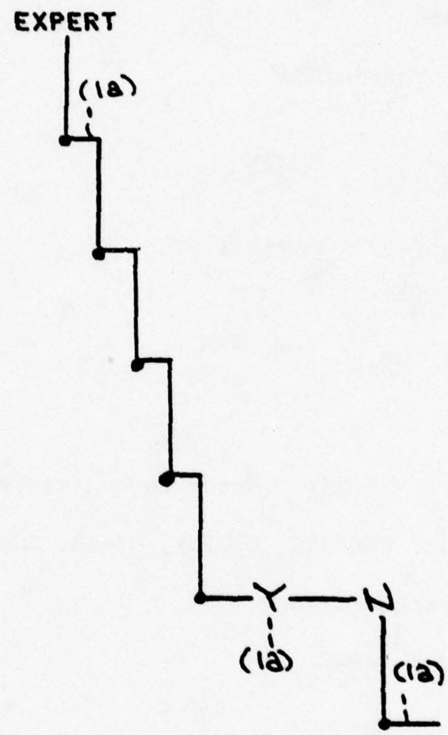
Cancels: None

Input arguments: TYPE — 0 non-demonstration mode; 1 demonstration mode

Common variables: None

Files created/changed: None

Files referenced: None



SET MEFIRST TO 0

CALL RANK

SET SAYTHIS TO 0

CALL PICKY

DOES TYPE = 0?

CALL GLIB

EXPERT FLOWCHART (SHEET 1 OF 1)

RANK

Description: Orders applicable messsages into a prioritized list.

Entry point: RANK

Classification: Subroutine

Period: None

Language: F

Activated/called by: EXPERT

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: MEFIRST — array to contain prioritized list

Common variables: CTEMERG, CTOTHR, CTNGR, ACGYRO, CTDOWN, CTATHT

Files created/changed: None

Files referenced: None

RANK

IS CTEMERG SET TO -1?

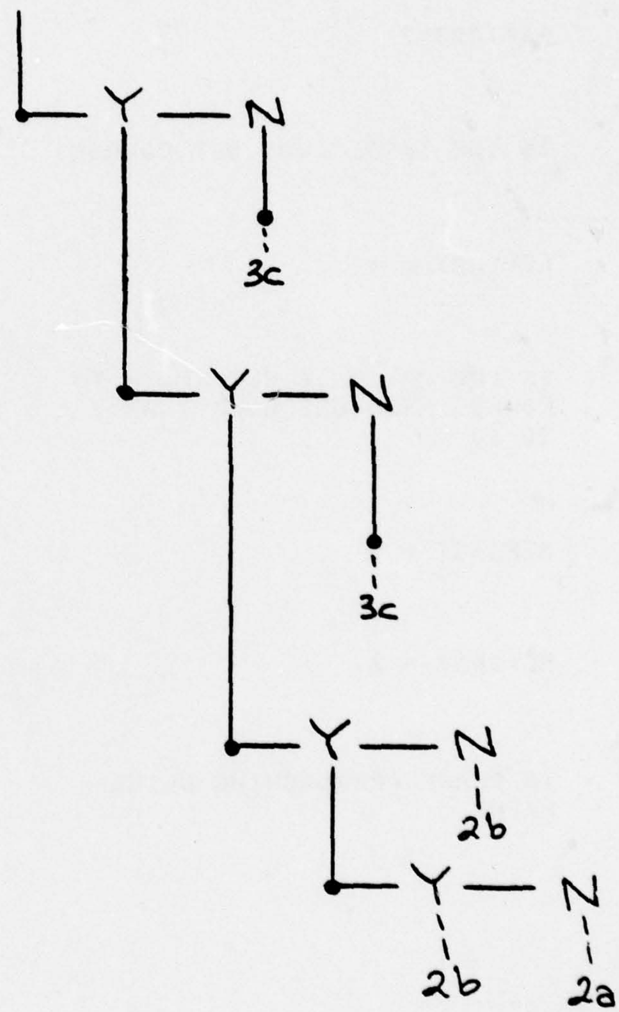
MEFIRST=1 (BASED ON GCA PRIORITY LIST - 1,2 AND 3 ARE EMERGENCIES)

IS CTOTHR SET TO -1?

MEFIRST=4 (SOMETHING MUST BE SAID NOW, INCLUDES HAND-OFFS, OVER LANDING THRESHOLD ETC. 4,8 ON PRIORITY LIST)

HAS "BEGIN DESCENT" BEEN GIVEN?

HAS "OVER LANDING THRESHOLD" BEEN GIVEN? (DOES CTOLT = .TRUE.?)



RANK FLOWCHART (SHEET 1 OF 3)

IS THE TARGET WITHIN .1 MILE OF A RANGE MARK?

MEFIRST=7

IS THE TARGET WAY OFF COURSE?

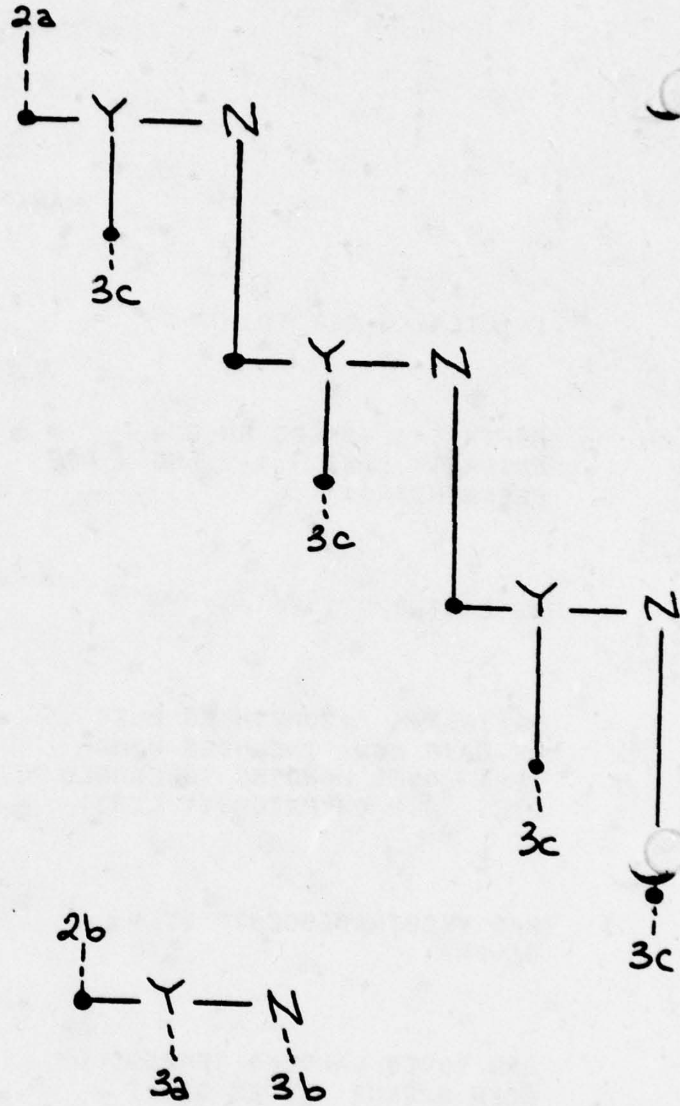
MEFIRST = 9

IS THE RATIO OF GLIDEPATH TO COURSE MESSAGES MORE THAN 3 TO 1?

MEFIRST = 9

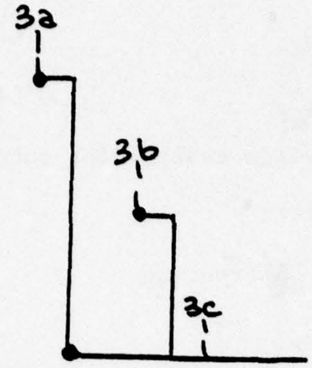
MEFIRST = 6

IS PLANE APPROACHING GLIDE-PATH?



RANK FLOWCHART (SHEET 2 OF 3)

MEFIRST = 11



MEFIRST = 0 (NO MESSAGE HAS PRIORITY. PICKY WILL FIGURE OUT SOMETHING)

RETURN

RANK FLOWCHART (SHEET 3 OF 3)

PICKY

Description: Builds message for output to GLIB.

Entry point: PICKY

Classification: Subroutine

Period: None

Language: F

Activated/called by: EXPERT

Cancelled by: N/A

Activates/calls: HEADING, NOGYRO, CSOVER, WINDY, DECK, BEATIT, GTREND, CTREND, HOWHIGH, HOWFAR

Cancels: None

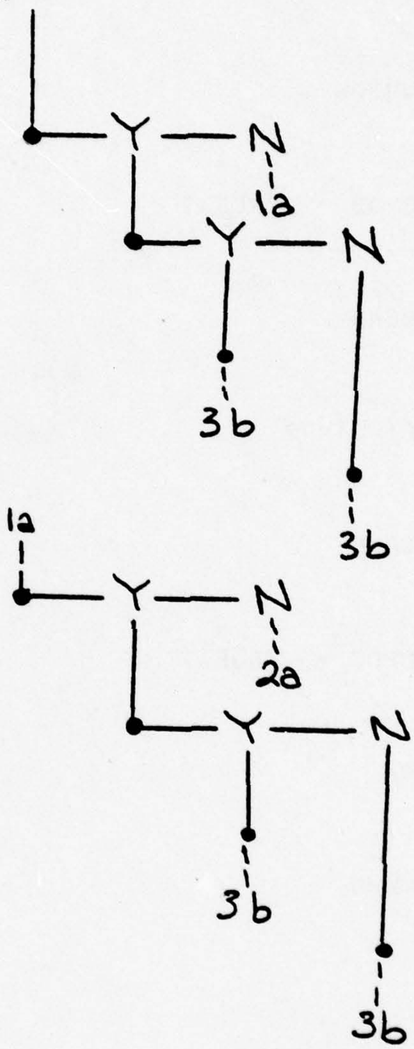
Input arguments: MEFIRST — prioritized list
SAYTHIS — array for Votrax output through VSCON

Common variables: CTEMERG, CTOTHR

Files created/changed: None

Files referenced: None

PICKY



DOES MEFIRST=1?
(IS IT AN EMERGENCY?)

IS IT A LOW ALTITUDE ALERT?

CALL MUNDANE (0,SAYTHIS)

CALL BEATIT

DOES MEFIRST=4? (IS IT
SOMETHING THAT MUST BE SAID?)

IS IT AT DECISION HEIGHT?

CALL DECK

CALL MUNDANE (1,SAYTHIS)

PICKY FLOWCHART (SHEET 1 OF 3)

DOES MEFIRST=6?

CALL HOWHIGH

DOES CTGPOS = .TRUE.?

CALL GTREND

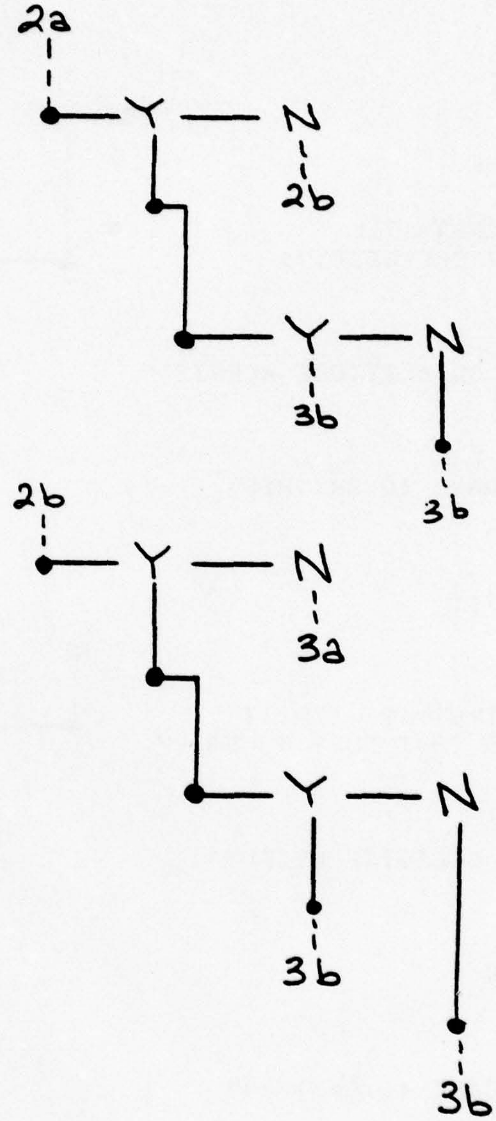
DOES MEFIRST=9?

CALL HOWFAR

DOES CTCPOS = .TRUE.?

CALL TURN

CALL CTREND



PICKY FLOWCHART (SHEET 2 OF 3)

DOES MEFIRST=7?

CALL RANGE

DOES MEFIRST=11?

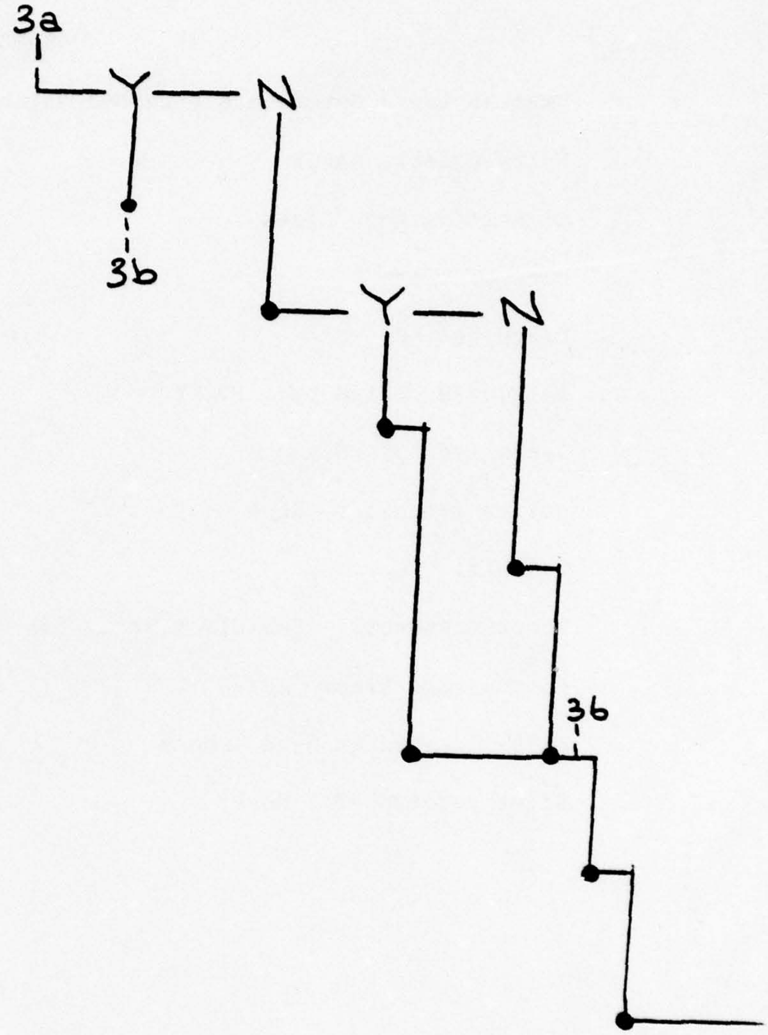
CALL AN APPROACHING GLIDE-
PATH ROUTINE

CALL A TIME FILLER ROUTINE
TO CONFORM TO THE 5 SECOND
RULE

TELL COMMON WHAT MESSAGE IS
SENT-(UPDATE COMMON BLOCK)

CALL CSOVER

RETURN



PICKY FLOWCHART (SHEET 3 OF 3)

RANGE

Description: Determines when and which range call should be given.

Entry point: RANGE

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — array for VOTRAX output via GLIB

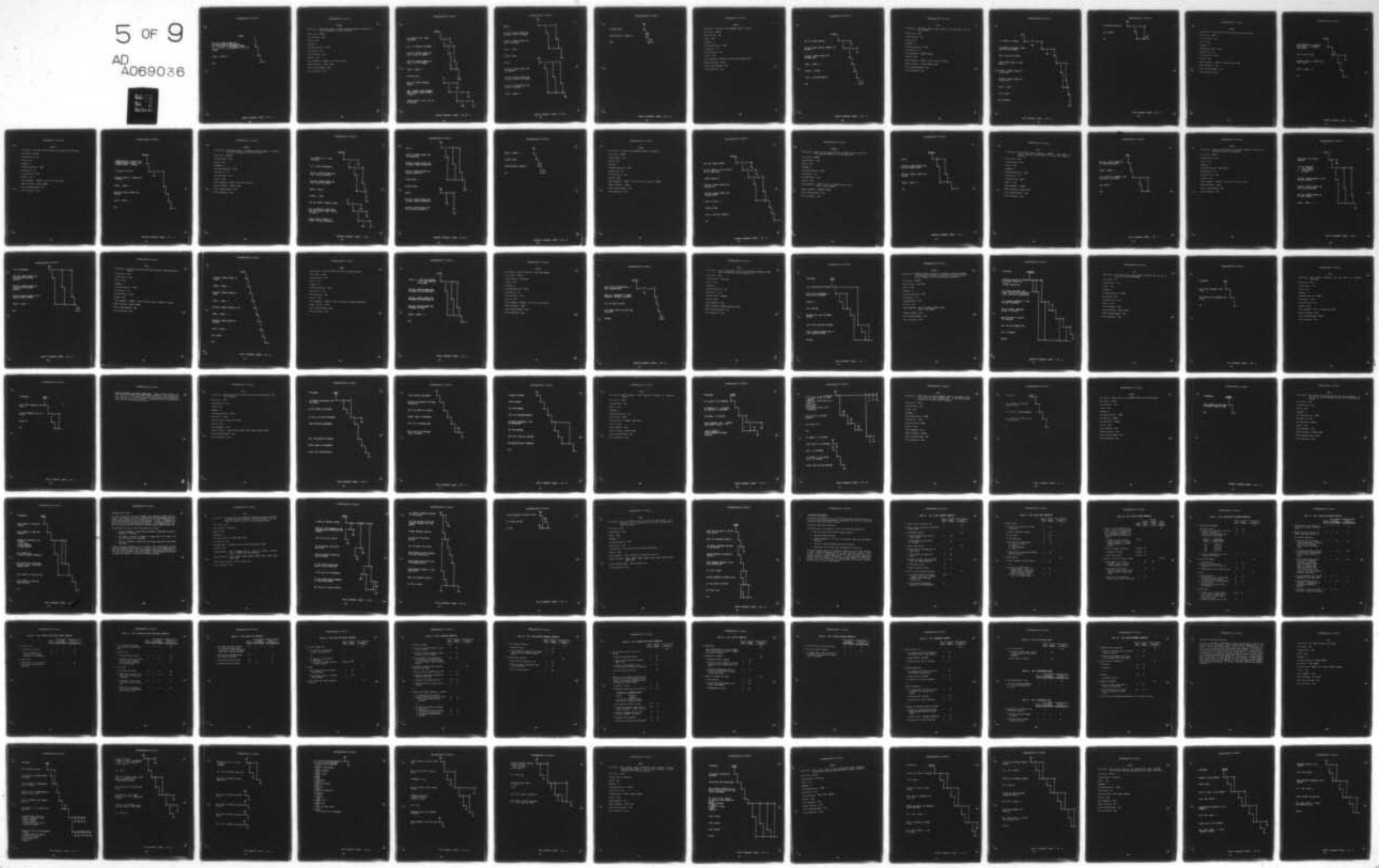
Common variables: ACRNG

Files created/changed: None

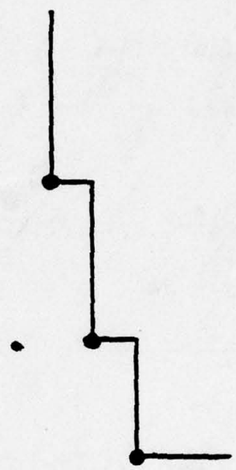
Files referenced: None

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

5 OF 9
AD
A069036



RANGE



SAYTHIS (INDEX)=INDEX OF
THE PHRASE WHICH CORRESPONDS
TO "'X' MILES TO TOUCHDOWN" WHERE
X = 1 TO 8. (THE NUMBER IS HELD IN
CTRNG)

INDEX = INDEX + 1

DIE

RANGE FLOWCHART (SHEET 1 OF 1)

HOWFAR

Description: Determines whether a course position message is required and adds the message to the array for GLIB.

Entry point: HOWFAR

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

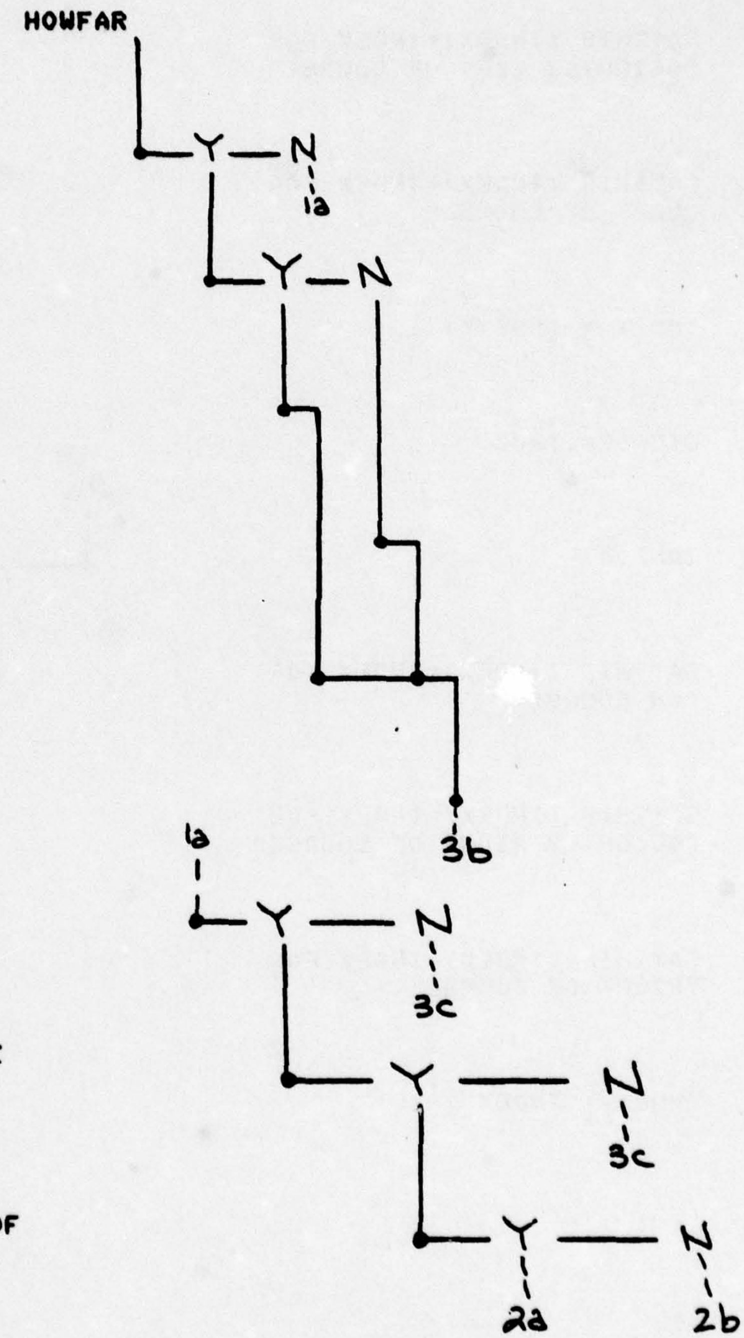
Cancels: None

Input arguments: SAYTHIS — the array for GLIB

Common variables: CTOPS, CTCRP

Files created/changed: None

Files referenced: None



IS AIRCRAFT IN A "WELL" POSITION?

IS IT TO RIGHT OF COURSE?

SAYTHIS (INDEX)=INDEX FOR "WELL RIGHT OF COURSE"

SAYTHIS (INDEX)=INDEX FOR "WELL LEFT OF COURSE"

INDEX = INDEX + 1

CTCPOS=.TRUE.

HAS THE TARGET CHANGED ZONES?

HAS A COURSE TREND MESSAGE BEEN GIVEN? (DOES CTCPOS= .FALSE.?)

TARGET MOVED TO THE LEFT OF ON COURSE?

HOWFAR FLOWCHART (SHEET 1 OF 3)

ZONE #

SAYTHIS (INDEX)=INDEX FOR
"SLIGHTLY LEFT OF COURSE"

SAYTHIS (INDEX)=INDEX FOR
"LEFT OF COURSE"

INDEX = INDEX + 1

CTCPOS=.TRUE.

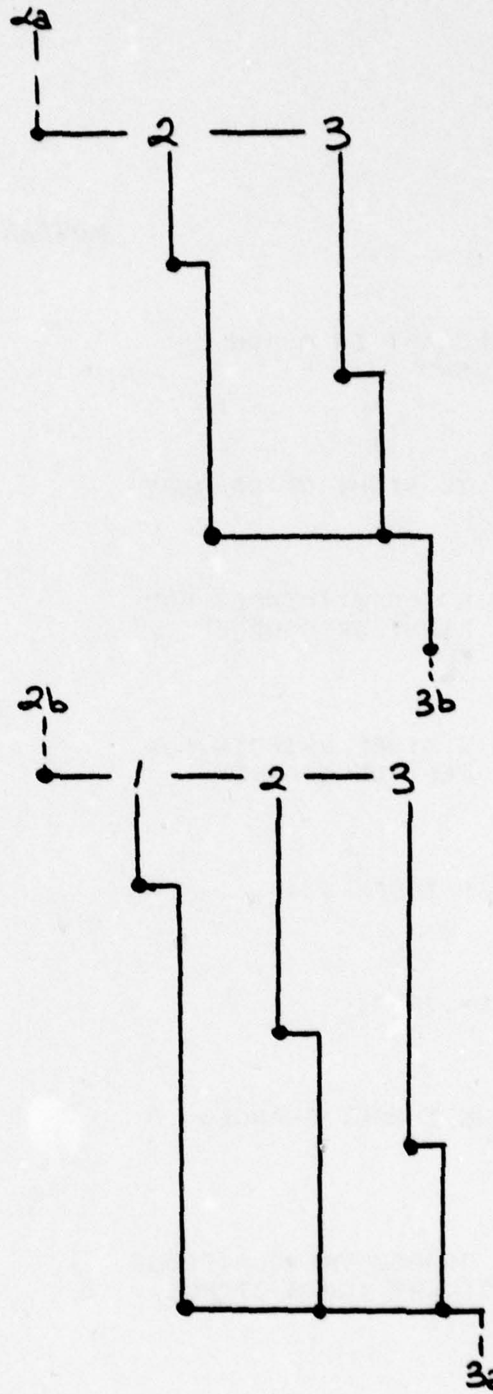
ZONE #

SAYTHIS (INDEX)=INDEX FOR
"ON COURSE"

SAYTHIS (INDEX)=INDEX FOR
"SLIGHTLY RIGHT OF COURSE"

SAYTHIS (INDEX)=INDEX FOR
"RIGHT OF COURSE"

INDEX = INDEX + 1

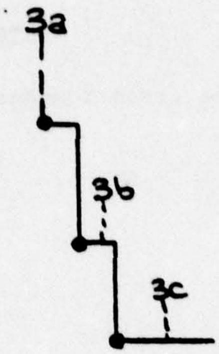


HOWFAR FLOWCHART (SHEET 2 OF 3)

CTCPOS=.TRUE.

CTCRF=SAYTHIS (INDEX-1)

DIE



HOWFAR FLOWCHART (SHEET 3 OF 3)

CTREND

Description: Calculates trend message needed for course.

Entry point: CTREND

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — array for GLIB VOTRAX output

Common variables: CTCPOS

Files created/changed: None

Files referenced: None

CTREND

HAS THE TARGET MOVED?

IS THE TARGET MOVING TOWARD "ON COURSE"?

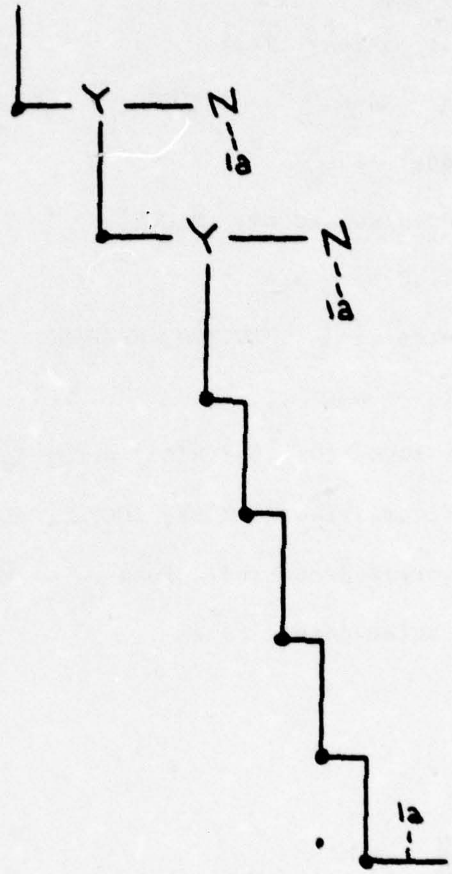
SAYTHIS (INDEX)=INDEX FOR "CORRECTING"

INDEX = INDEX + 1

CTCPOS = .FALSE.

CTCRT = SAYTHIS(INDEX-1)

DIE



CTREND FLOWCHART (SHEET 1 OF 1)

TURN

Description: Determines whether target needs to turn and adds it to the array SAYTHIS.

Entry point: TURN

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: NOGYRO, HEADING

Cancels: None

Input arguments: SAYTHIS — array for Votrax output

Common variables: ACOFF, ACGYRO, ACHDG

Files created/changed: None

Files referenced: None

TURN

IS TARGET ON COURSE?

IS TARGET IN SLIGHTLY ZONE, PARALLELING COURSE?

RESET CLOCK FOR TURNS

DOES TARGET NEED TO TURN RIGHT?

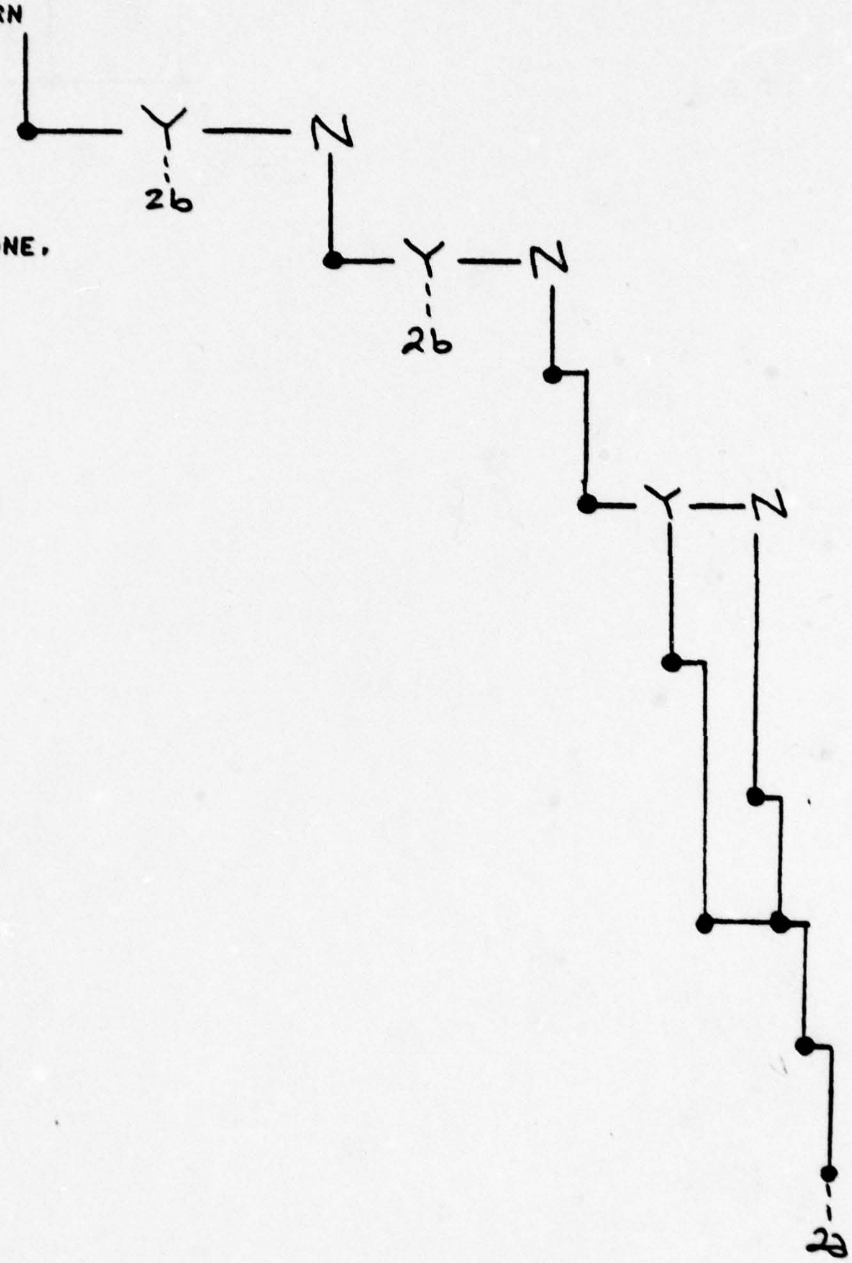
SAYTHIS (INDEX)=INDEX OF "TURN RIGHT"

SAYTHIS (INDEX)=INDEX OF "TURN LEFT"

INDEX = INDEX + 1

START CLOCK

CALL HEADING

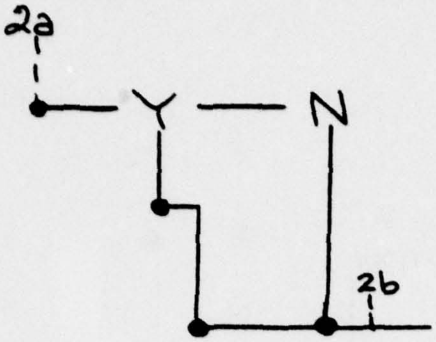


TURN FLOWCHART (SHEET 1 OF 2)

IS NOGYRO FLAG SET?

CALL NOGYRO

DIE



TURN FLOWCHART (SHEET 2 OF 2)

NOGYRO

Description: Starts and stops turns when a gyro failure occurs.

Entry point: NOGYRO

Classification: Task

Period: None

Language: ¯F

Activated/called by: TURN

Cancelled by: N/A

Activates/calls: None

Cancels: None

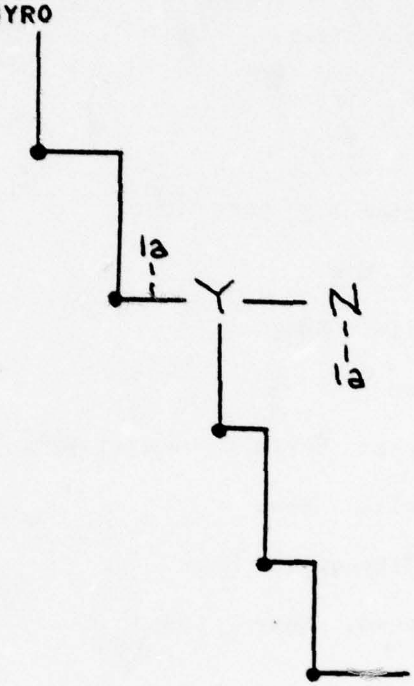
Input arguments: SAYTHIS — output array for GLIB

Common variables: None

Files created/changed: None

Files referenced: None

NOGYRO



TIME=NEWHEAD/1.5 (TIME TO STOP TURN = 1.5 DEGREES PER SECOND)

TIME TO STOP TURN?

SAYTHIS (INDEX) = INDEX FOR "STOP TURN"

INDEX = INDEX + 1

DIE

NOGYRO FLOWCHART (SHEET 1 OF 1)

HEADING

Description: Calculates heading from plane's direction and wind data.

Entry point: HEADING

Classification: Task

Period: None

Language: F

Activated/called by: TURN

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — array for Votrax output

Common variables: ENWSP, ENWHDG

Files created/changed: None

Files referenced: None

HEADING

NEWHEAD=ARCTAN ((ENWSP* SIN (ENWHDG-ACHDG))/ENWSP * COS (ENWHDG-ACHDG) + ACGSPD

IS NOGYRO FLAG SET?

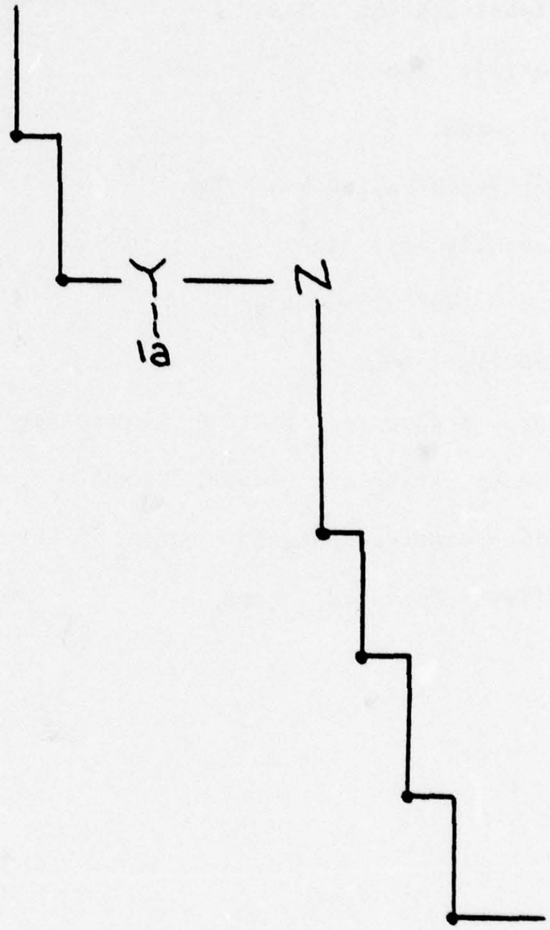
SAYTHIS (INDEX) = INDEX FOR "HEADING"

INDEX = INDEX + 1

SAYTHIS (INDEX)=INDEX FOR NEWHEAD

INDEX = INDEX + 1

DIE



HEADING FLOWCHART (SHEET 1 OF 1)

HOWHIGH

Description: Determines whether a glidepath position message is required and adds the message to the array for GLIB.

Entry point: HOWHIGH

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — the array for GLIB

Common variables: CTGPOS, CTGPP

Files created/changed: None

Files referenced: None

IS AIRCRAFT IN A "WELL" POSITION?

IS IT ABOVE GLIDEPATH?

SAYTHIS (INDEX)=INDEX FOR "WELL ABOVE GLIDEPATH"

SAYTHIS (INDEX)=INDEX FOR "WELL BELOW GLIDEPATH"

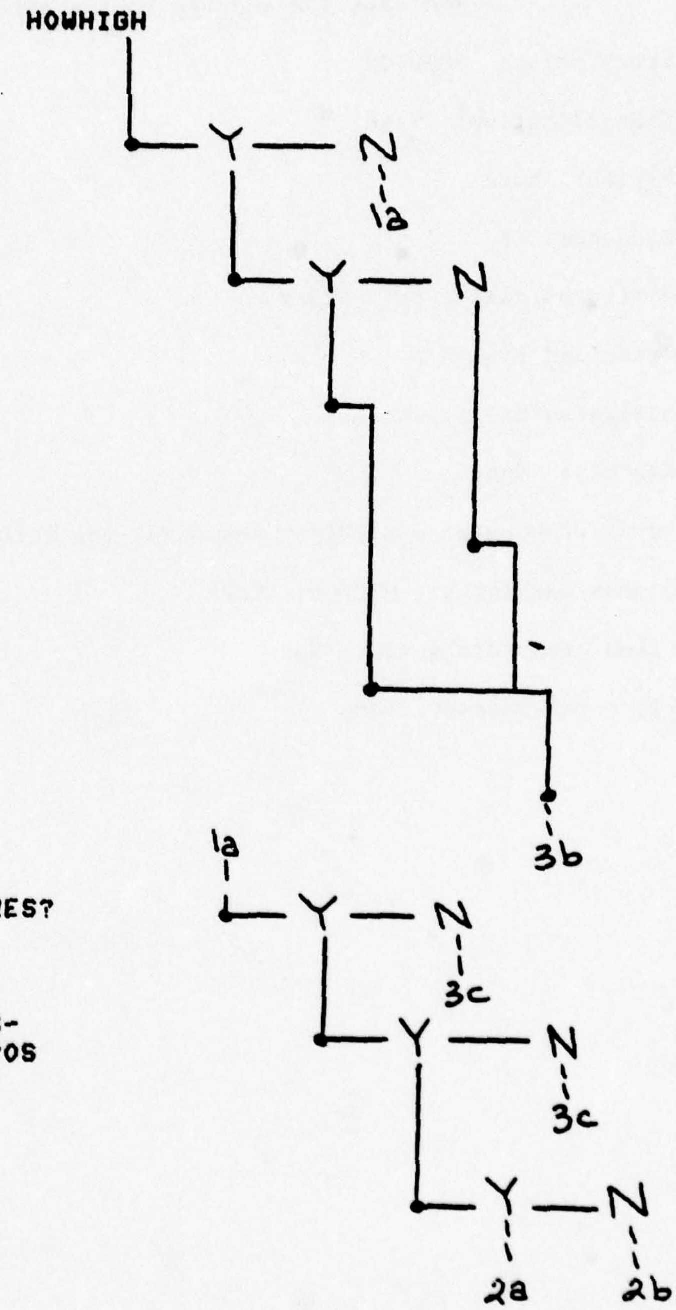
INDEX = INDEX + 1

CTGPOS = .TRUE

HAS THE TARGET CHANGED ZONES?

HAS A GLIDEPATH TREND MESSAGE BEEN GIVEN? (DOES CTGPOS =.FALSE.?)

TARGET MOVED UPWARDS OF SLIGHTLY BELOW GLIDEPATH ?



HOWHIGH FLOWCHART (SHEET 1 OF 3)

ZONE #

SAYTHIS (INDEX)=INDEX FOR
"ON GLIDEPATH"

SAYTHIS (INDEX)=INDEX FOR
"SLIGHTLY ABOVE GLIDEPATH"

SAYTHIS (INDEX)=INDEX FOR
"ABOVE GLIDEPATH"

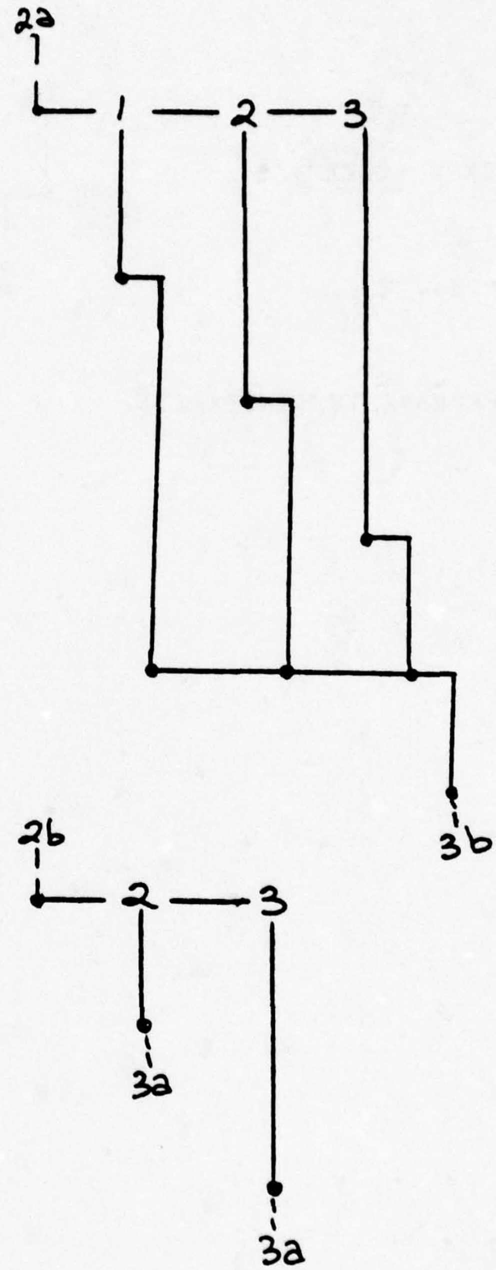
INDEX=INDEX + 1

CTGPOS=.TRUE.

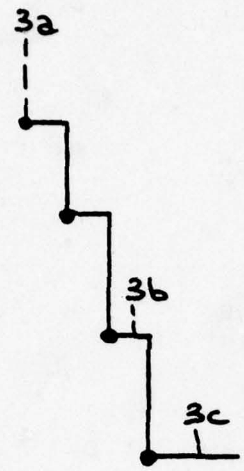
ZONE #

SAYTHIS (INDEX)=INDEX FOR
"SLIGHTLY BELOW GLIDEPATH"

SAYTHIS (INDEX)=INDEX FOR
"WILL BELOW GLIDEPATH"



HOWHIGH FLOWCHART (SHEET 2 OF 3)



INDEX = INDEX + 1

CTGPOS=.TRUE.

CTGPP=SAYTHIS (INDEX-1)

DIE

HOWHIGH FLOWCHART (SHEET 3 OF 3)

GTREND

Description: Calculates trend message needed for glidepath.

Entry point: GTREND

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — array for GLIB to output to VOTRAX

Common variables: CTGPOS

Files created/changed: None

Files referenced: None

HAS THE TARGET MOVED?

IS THE TARGET 1/3 OF THE WAY
INTO A NEW ZONE?

TARGET MOVED UP?

SAYTHIS (INDEX)=INDEX FOR
"COMING UP"

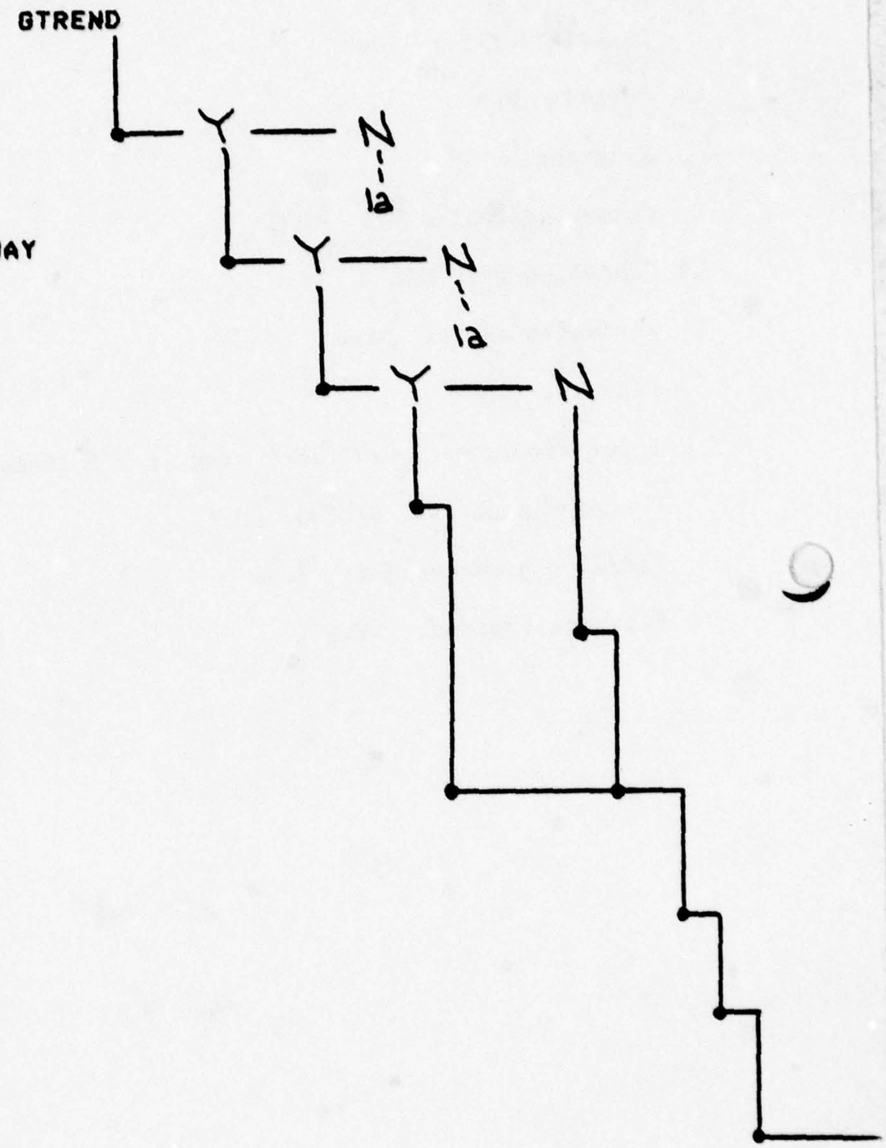
SAYTHIS (INDEX)=INDEX FOR
"COMING DOWN"

INDEX = INDEX + 1

CTGPOS=.FALSE.

CTGPT = SAYTHIS (INDEX-1)

DIE



GTREND FLOWCHART (SHEET 1 OF 1)

MUNDANE

Description: Takes care of phrases which are the same in every run such as "over landing threshold" or low altitude alert.

Entry point: MUNDANE

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

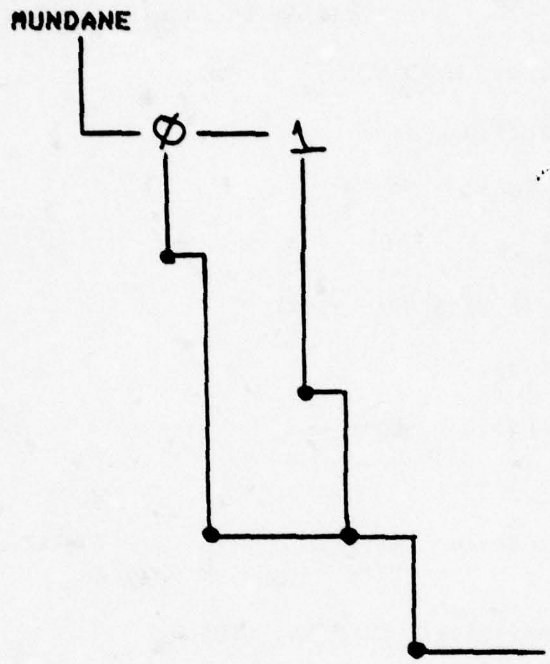
Cancels: None

Input arguments: SAYTHIS — array for VOTRAX output to GLIB
TYPE — type of message

Common variables: CTEMERG, CTOTHR

Files created/changed: None

Files referenced: None



TYPE N

SAYTHIS (INDEX)=INDEX FOR
"LOW ALTITUDE ALERT"

SAYTHIS (INDEX)=INDEX FOR
CTOTHR

INDEX = INDEX + 1

DIE

MUNDANE FLOWCHART (SHEET 1 OF 1)

DECK

Description: Puts decision height message in SAYTHIS. Calls BEATIT if waveoff necessary because target is too far off course or glidepath.

Entry point: DECK

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: BEATIT

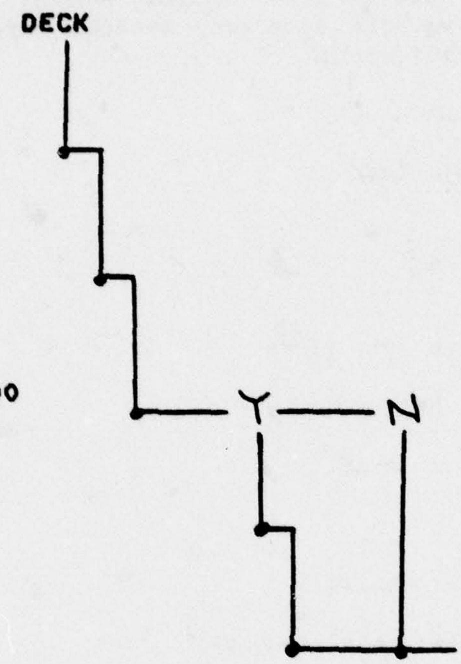
Cancels: None

Input arguments: SAYTHIS

Common variables: ACALT, ACOFF

Files created/changed: None

Files referenced: None



SAYTHIS (INDEX)=INDEX OF "AT DECISION HEIGHT"

INDEX = INDEX + 1

IS A WAVEOFF IN ORDER? (TOO HIGH/LOW, LEFT/RIGHT)

CALL BEATIT

DIE

DECK FLOWCHART (SHEET 1 OF 1)

BEATIT

Description: Handles both mandatory and optional waveoffs, selecting the appropriate type for each approach.

Entry point: BEATIT

Classification: Task

Period: None

Language: F

Activated/called by: DECK, PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: SAYTHIS — the array for output to GLIB

Common variables: CTCLR

Files created/changed: None

Files referenced: None

MANDATORY OR OPTIONAL
WAVEOFF?

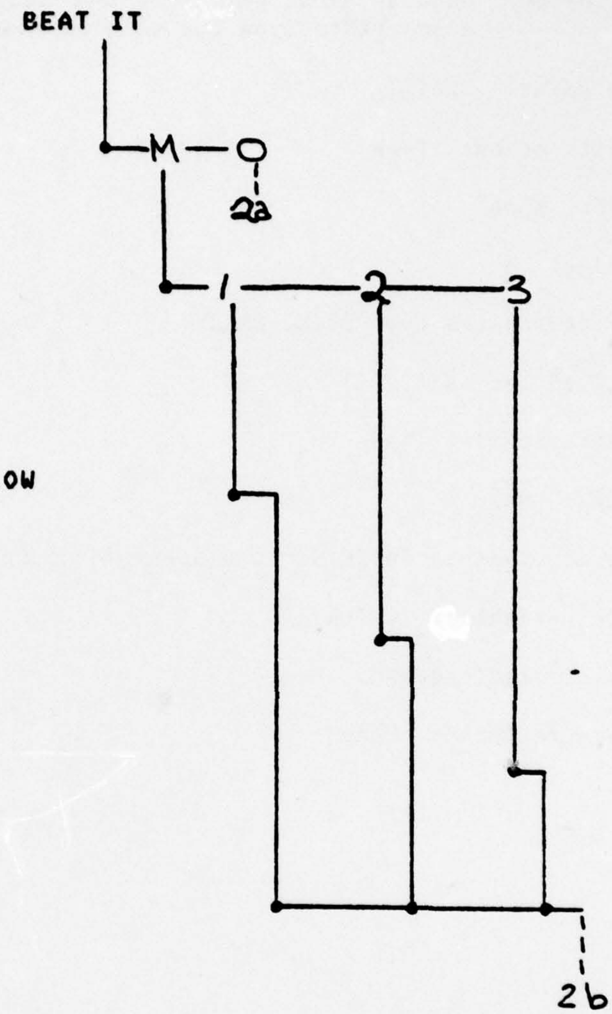
TYPE OF APPROACH
1   LOW APPROACH
2   TOUCH AND GO
3   TO LAND

SAYTHIS (INDEX)=INDEX TO LOW
APPROACH WAVEOFF

SAYTHIS (INDEX)=INDEX FOR
TOUCH AND GO WAVEOFF

SAYTHIS (INDEX)=INDEX FOR
TO LAND WAVEOFF

INDEX = INDEX + 1



BEATIT FLOWCHART (SHEET 1 OF 2)

TYPE OF APPROACH

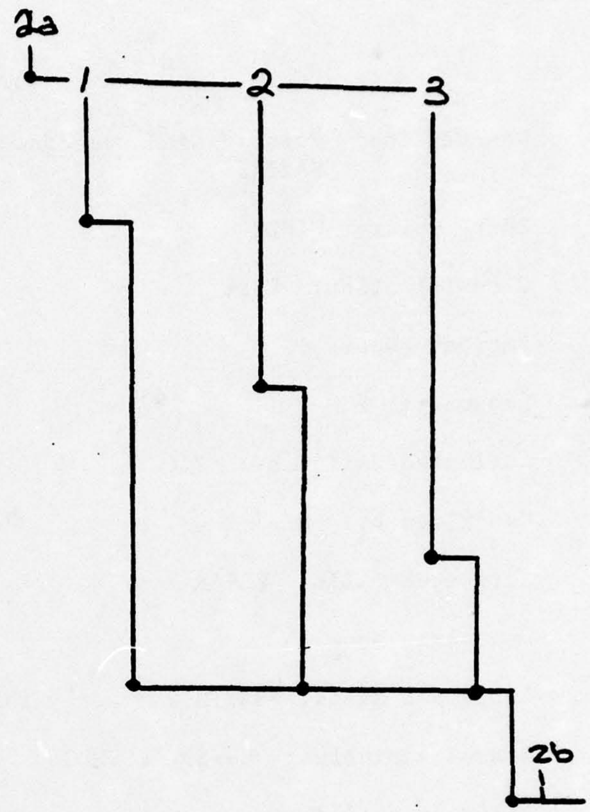
SAYTHIS (INDEX)=INDEX FOR
LOW APPROACH OPTIONAL
WAVEOFF

SAYTHIS (INDEX)=INDEX FOR
TOUCH AND GO OPTIONAL
WAVEOFF

SAYTHIS (INDEX)=INDEX FOR TO
LAND OPTIONAL WAVEOFF

INDEX = INDEX + 1

DIE



BEATIT FLOWCHART (SHEET 2 OF 2)

WINDY

Description: Puts in wind speed and direction and adds completed message to SAYTHIS.

Entry point: WINDY

Classification: Task

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: CLEAR

Cancels: None

Input arguments: SAYTHIS — array for GLIB to send to VOTRAX for output

Common variables: ENWSP, ENWHDG

Files created/changed: None

Files referenced: None

WINDY

SAYTHIS (INDEX)=INDEX OF
"WIND"

INDEX = INDEX + 1

SAYTHIS (INDEX)=INDEX OF
ENWSP

INDEX = INDEX + 1

SAYTHIS (INDEX)=INDEX OF "AT"

INDEX = INDEX + 1

SAYTHIS (INDEX)=INDEX OF
"ENWHDG"

INDEX = INDEX + 1

CALL CLEAR

DIE

WINDY FLOWCHART (SHEET 1 OF 1)

CLEAR

Description: Puts into SAYTHIS the type of clearance desired.

Entry point: CLEAR

Classification: Task

Period: None

Language: F

Activated/called by: None

Cancelled by: N/A

Activates/calls: None

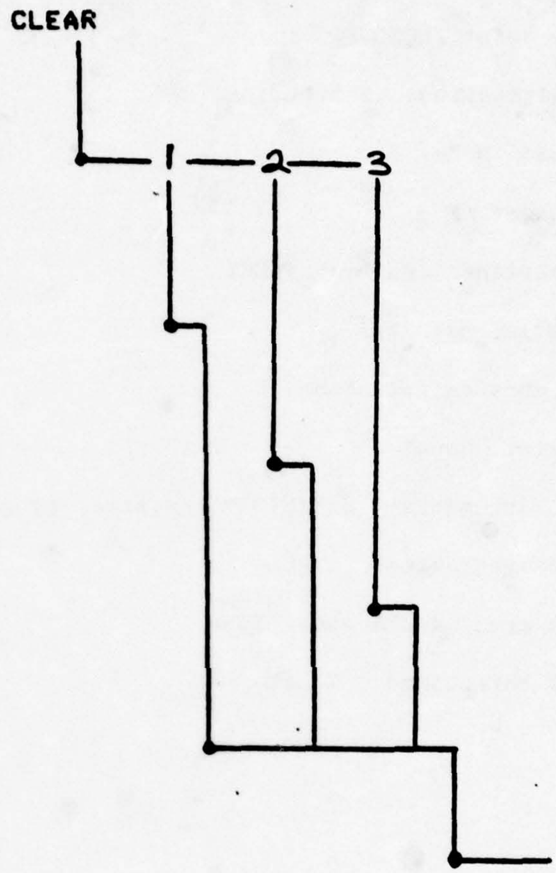
Cancels: None

Input arguments: SAYTHIS — array for output to VOTRAX through GLIB

Common variables: CTCLR

Files created/changed: None

Files referenced: None



CTCLR = 1 FOR LOW APPROACH
2 FOR TOUCH AND GO
3 TO LAND

SAYTHIS (INDEX)=INDEX FOR
"CLEARED FOR LOW APPROACH"

SAYTHIS (INDEX)=INDEX FOR
"CLEARED FOR TOUCH AND GO"

SAYTHIS (INDEX)=INDEX FOR
"CLEARED TO LAND"

INDEX = INDEX + 1

DIE

CLEAR FLOWCHART (SHEET 1 OF 1)

CSOVER

Description: Adds call sign and "over" when needed.

Entry point: CSOVER

Classification: Subroutine

Period: None

Language: F

Activated/called by: PICKY

Cancelled by: N/A

Activates/calls: None

Cancels: None

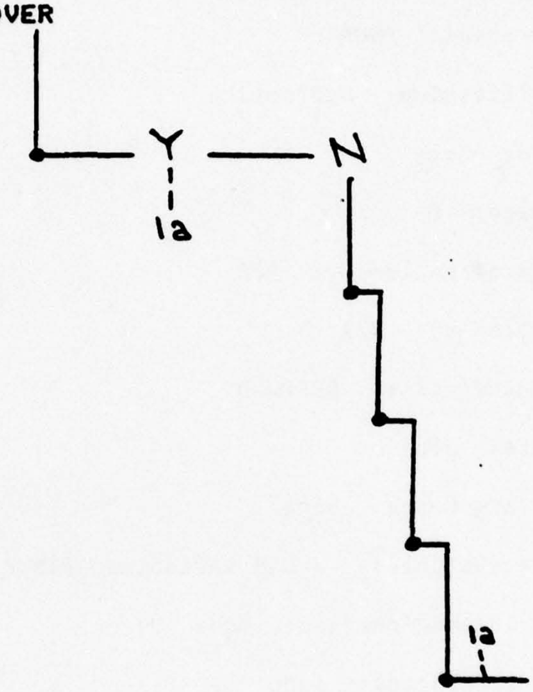
Input arguments: SAYTHIS — the array to go to GLIB

Common variables: CTACK

Files created/changed: None

Files referenced: None

CSOVER



HAS "DO NOT ACKNOWLEDGE..." BEEN TRANSMITTED?

MOVE ALL ARGUMENTS IN ARRAY DOWN BY 1 SO ARG 1 IS EMPTY

PUT C/S ARG# IN ARG 1

PUT "OVER" ARG # IN LAST ARG POSITION

RETURN

CSOVER FLOWCHART (SHEET 1 OF 1)

PMON

Description: This routine updates the list of acceptable messsages in CTMSG after a recognition occurs in phase 2 or 3.

Entry point: PMON

Classification: Subroutine

Period: None

Language: F

Activated/called by: APE

Cancelled by: N/A

Activates/calls: RESPOND

Cancels: None

Input arguments: None

Common variables: SHUSH variables, PTBYE

Files created/changed: None

Files referenced: None

PROCEDURE **PMON**

DID A RECOGNITION OCCUR?

WAS IT IN THE QUEUE OF
ACCEPTABLE MESSAGES?

REMOVE IT

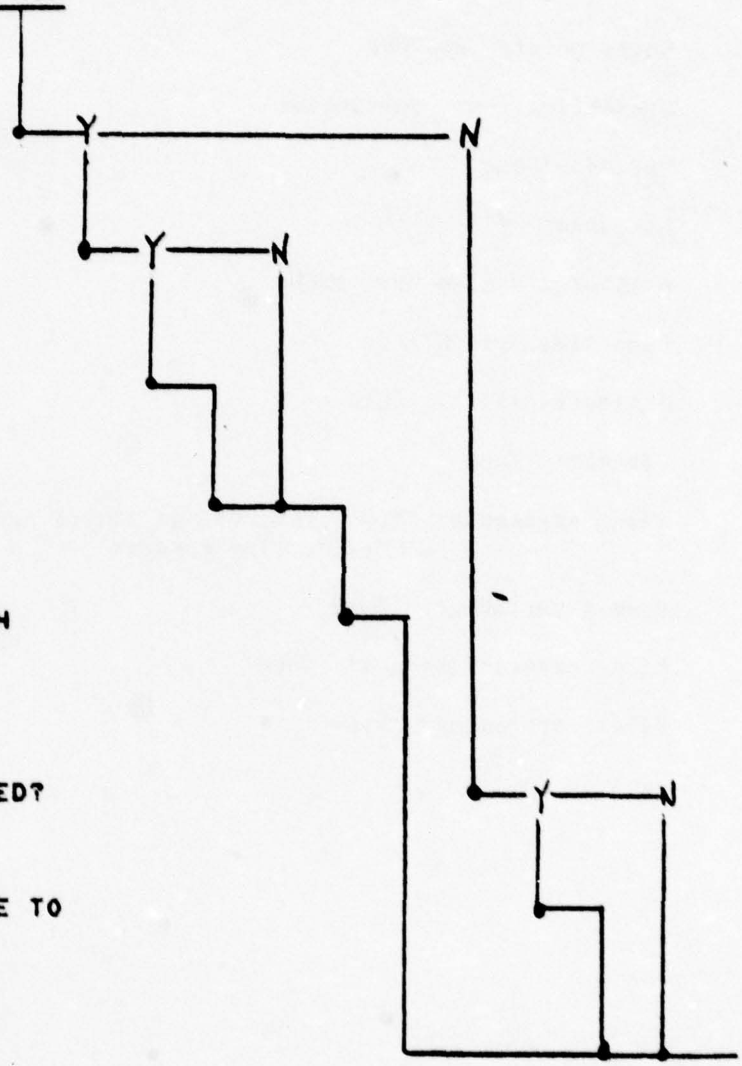
CALL RESPOND

RELEASE PMS LOCK ON SHUSH
BUFFER

HAVE PTBYE SECONDS ELAPSED?

START MISSED APPROACH DUE TO
LOST COMMUNICATIONS

RETURN



PMON FLOWCHART (SHEET 1 OF 1)

RESPOND

Description: RESPOND maintains the queue of acceptable controller messages associated with responses to the pilot or pattern controller. It also responds to the final controller as required.

Entry point: RESPOND

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMON

Cancelled by: N/A

Activates/calls: GLIB

Cancels: None

Input arguments: MSG — The message number spoken
ORIGIN — The speaker

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE RESPOND

BRANCH ON ORIGIN OF MESSAGE
1-PATTERN CONTROLLER
2-PILOT
3-FINAL CONTROLLER

SET PROPER RESPONSE INTO CTMSG, THE SET OF ACCEPTABLE FINAL CONTROLLER ADVISORIES

IS A VERBAL RESPONSE TO THIS ADVISORY REQUIRED?

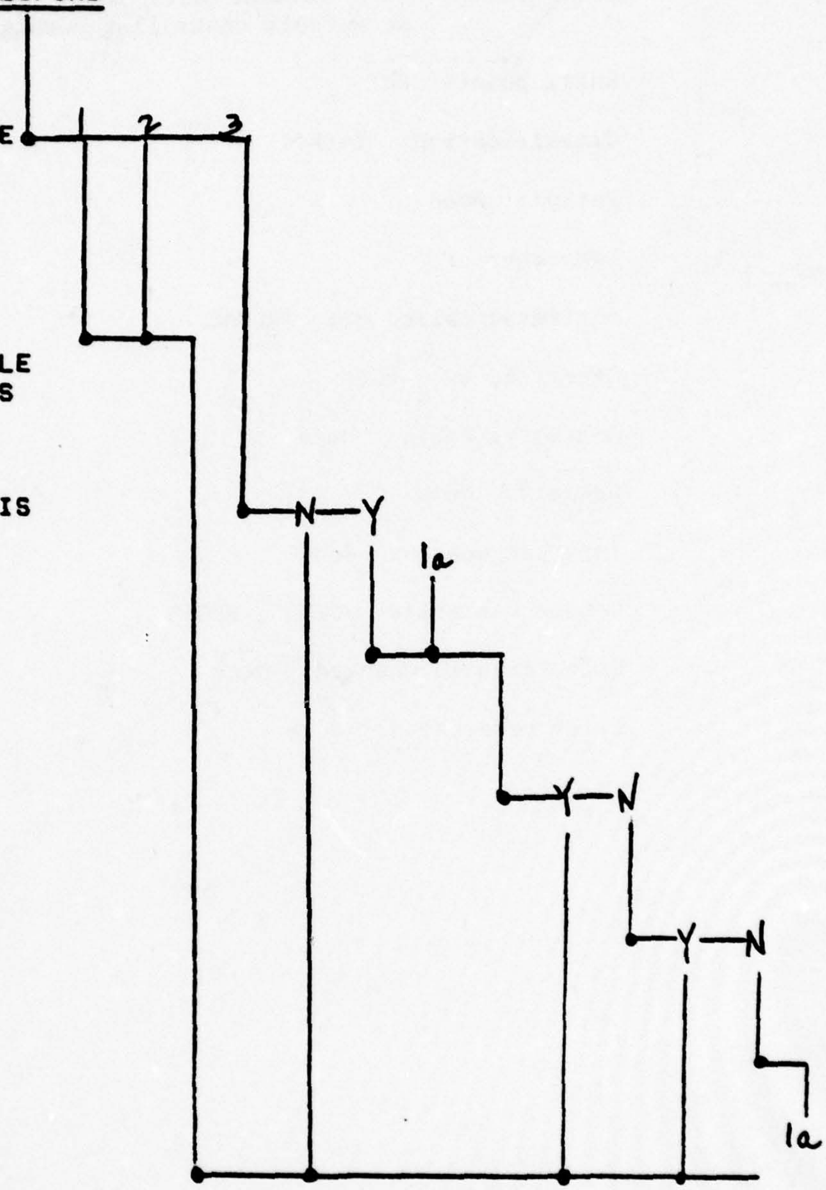
SELECT VERBAL RESPONSE, OUTPUT IT TO GLIB

WAS GLIB ABLE TO OUTPUT THE MESSAGE?

WAS THIS THE SECOND TRY?

WAIT 3 SECONDS

RETURN



RESPOND FLOWCHART (SHEET 1 OF 1)

PUT

Description: This routine sets range related advisories into the set of acceptable controller messages.

Entry point: PUT

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: PUT

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: CTMSG, PFQUE

Files created/changed: None

Files referenced: None

PROCEDURE PUT

COPY INPUT ARGUMENT ONTO
STACK

SET APPROPRIATE MESSAGE INTO
CTMSG

DIE

PUT FLOWCHART (SHEET 1 OF 1)

OMCK

Description: OMCK removes advisories from the queue of acceptable advisories.

Entry point: OMCK

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: OMCK

Activates/calls: None

Cancels: None

Input arguments: MSG — the index into PFQUE

Common variables: CTMSG

Files created/changed: RPLACT

Files referenced: None

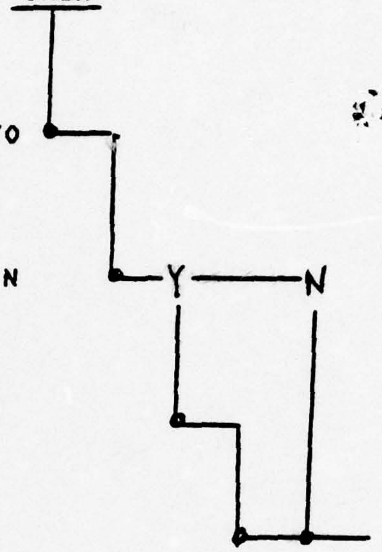
PROCEDURE OMCK

COPY INPUT ARGUMENT ONTO STACK

IS THIS MESSAGE STILL IN QUEUE?

REMOVE IT

DIE



OMCK FLOWCHART (SHEET 1 OF 1)

<u>Pattern Controller and Tower Simulations</u>. These routines simulate the activities of the other air traffic control personnel in the GCA environment. Entries in the problem description files specify such parameters as type of handoff and type of clearance. Program descriptions for the pattern controller and tower simulations follow.

FEED

Description: FEED simulates the activities of the pattern controller in the initial handoff.

Entry point: FEED

Classification: Task

Period: None

Language: F

Activated/called by: P23SUB

Cancelled by: FEED

Activates/calls: GLIB, PIN, TIMOUT

Cancels: None

Input arguments: None

Common variables: CTNGR, KYIC4, CTOTHR, CTREL, ENPSP, ENPHDG, PTAPR

Files created/changed: None

Files referenced: None

PROCEDURE

IS STUDENT MONITORING THE ICS PANEL?

BLINK FEEDER ICS BUTTON

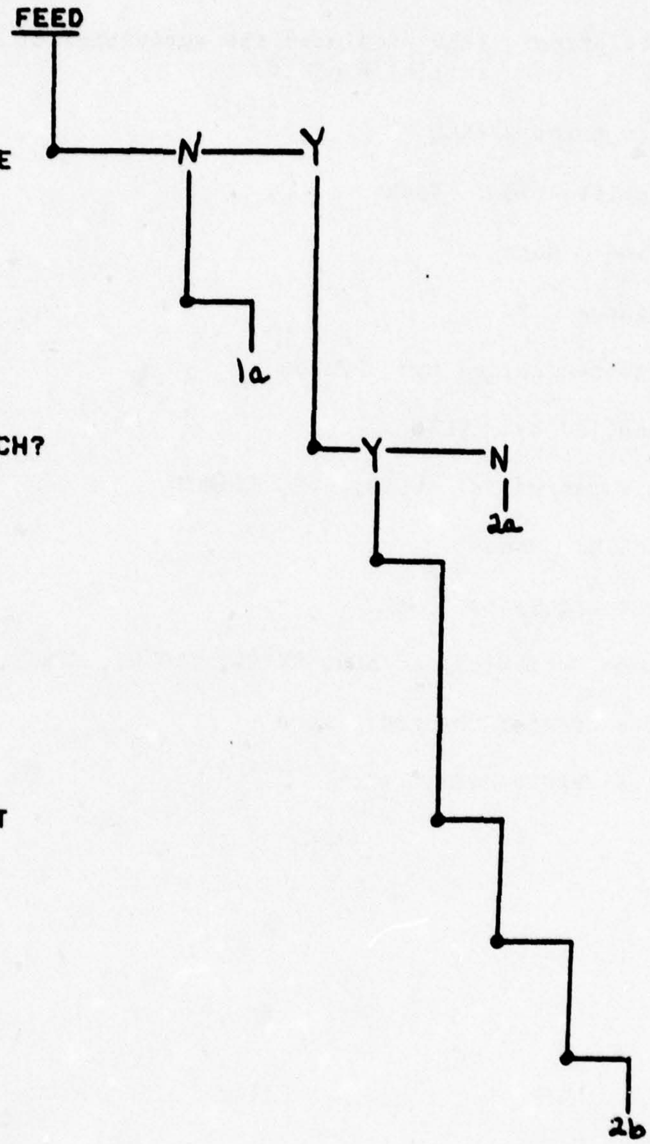
IS THIS A NO GYRO APPROACH?

ISSUE NO-GYRO ASSIGNMENT

WAIT FOR ROGER OR TIMEOUT

REPEAT ONCE IF NECESSARY

GIVE PILOT INSTRUCTIONS



FEED FLOWCHART (SHEET 1 OF 3)

ISSUE NORMAL ASSIGNMENT

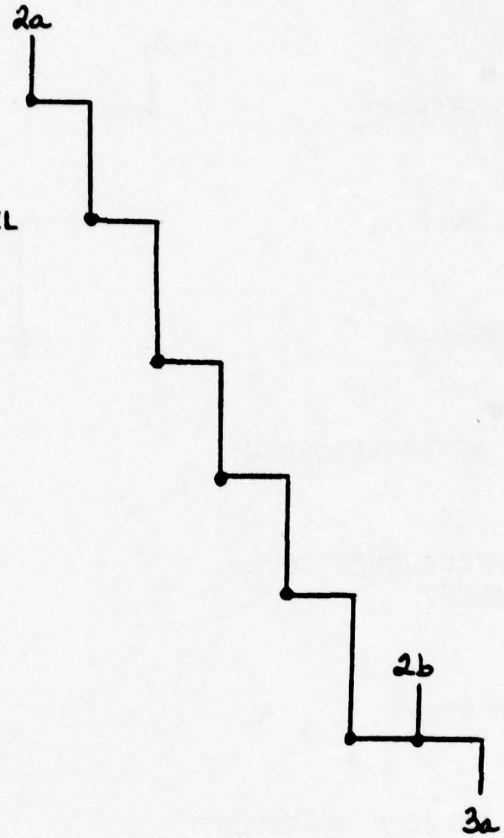
SELECT SUS MESSAGE FOR MODEL CONTROLLER

WAIT FOR ROGER OR TIMEOUT

REPEAT ONCE IF NECESSARY

GIVE PILOT INSTRUCTIONS

WAIT FOR PILOT RESPONSE ($VRO FINISHED)



FEED FLOWCHART (SHEET 2 OF 3)

RELEASE AIRCRAFT

ISSUE HANDOFF

SET SUS MESSAGE

WAIT FOR ACKNOWLEDGEMENT

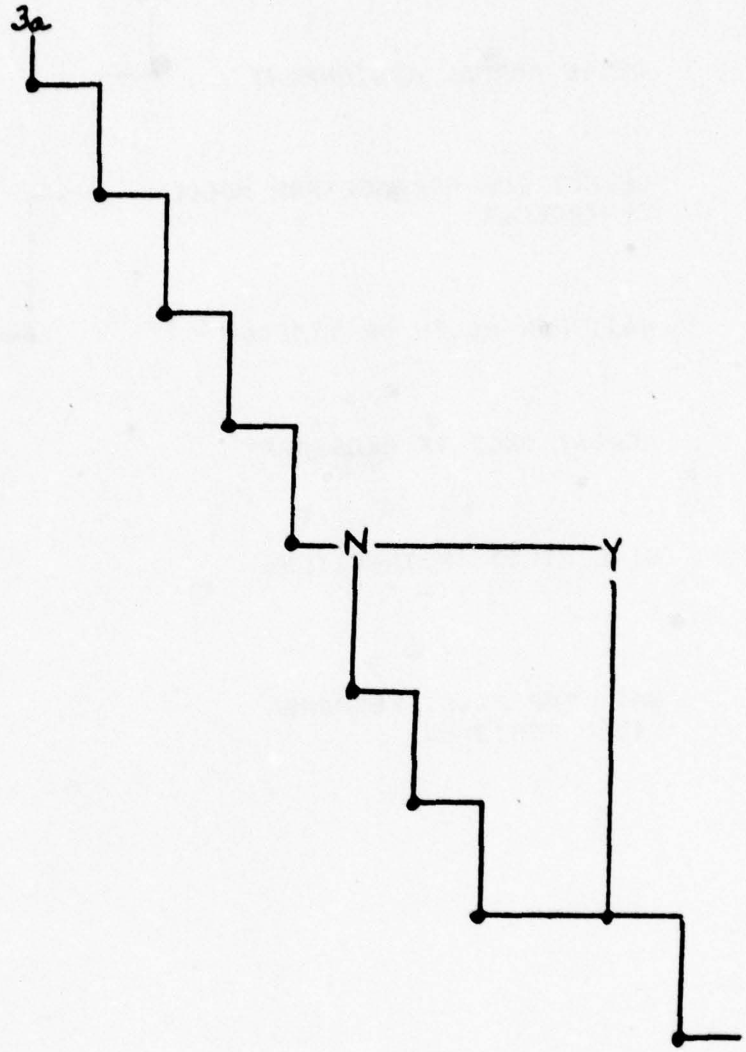
IS RADIO FREQUENCY TO BE RELINQUISHED?

SET SUS MESSAGE

WAIT FOR "GIVE ME" REQUEST

RELINQUISH RADIO FREQUENCY

DIE



FEED FLOWCHART (SHEET 3 OF 3)

TOWER

Description: TOWER simulates tower controller responses to clearance requests.

Entry point: TOWER

Classification: Task

Period: None

Language: F

Activated/called by: PIN

Cancelled by: TOWER

Activates/calls: IGNORE, CLRNC, WAVE

Cancels: None

Input arguments: None

Common variables: CTCLR, ACRNG

Files created/changed: None

Files referenced: None

PROCEDURE | TOWER

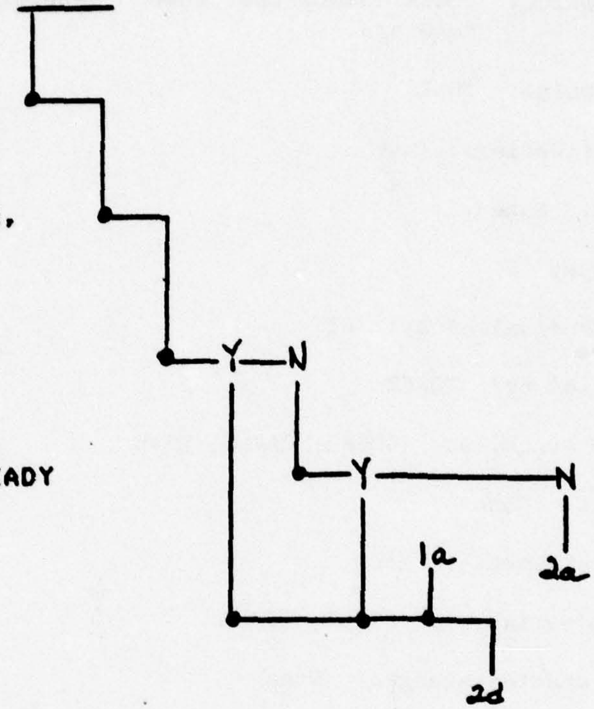
SET OUTIME TO 10 SECONDS

IF RANGE IS < = 1.5 MILES,
SET OUTIME TO 3 SECONDS

IS RANGE > 3.5 MILES?

HAS CLEARANCE, ETC. ALREADY
BEEN GIVEN (CTCLR = -1)?

IGNORE REQUEST:
START IGNORE IN OUTIME
SECONDS



TOWER FLOWCHART (SHEET 1 OF 2)

BRANCH ON TYPE OF CLEARANCE
1-CLEARANCE GIVEN AT FIRST
  REQUEST
2-CONTINUE, THEN CLEAR AT 2
  MILES
3-NOT GIVEN
4-WAVE OFF
5-CLEARANCE GIVEN, THEN
  CANCELLED

START CLEAR IN OUTIME
SECONDS

SET CTCLR TO -1

DIE

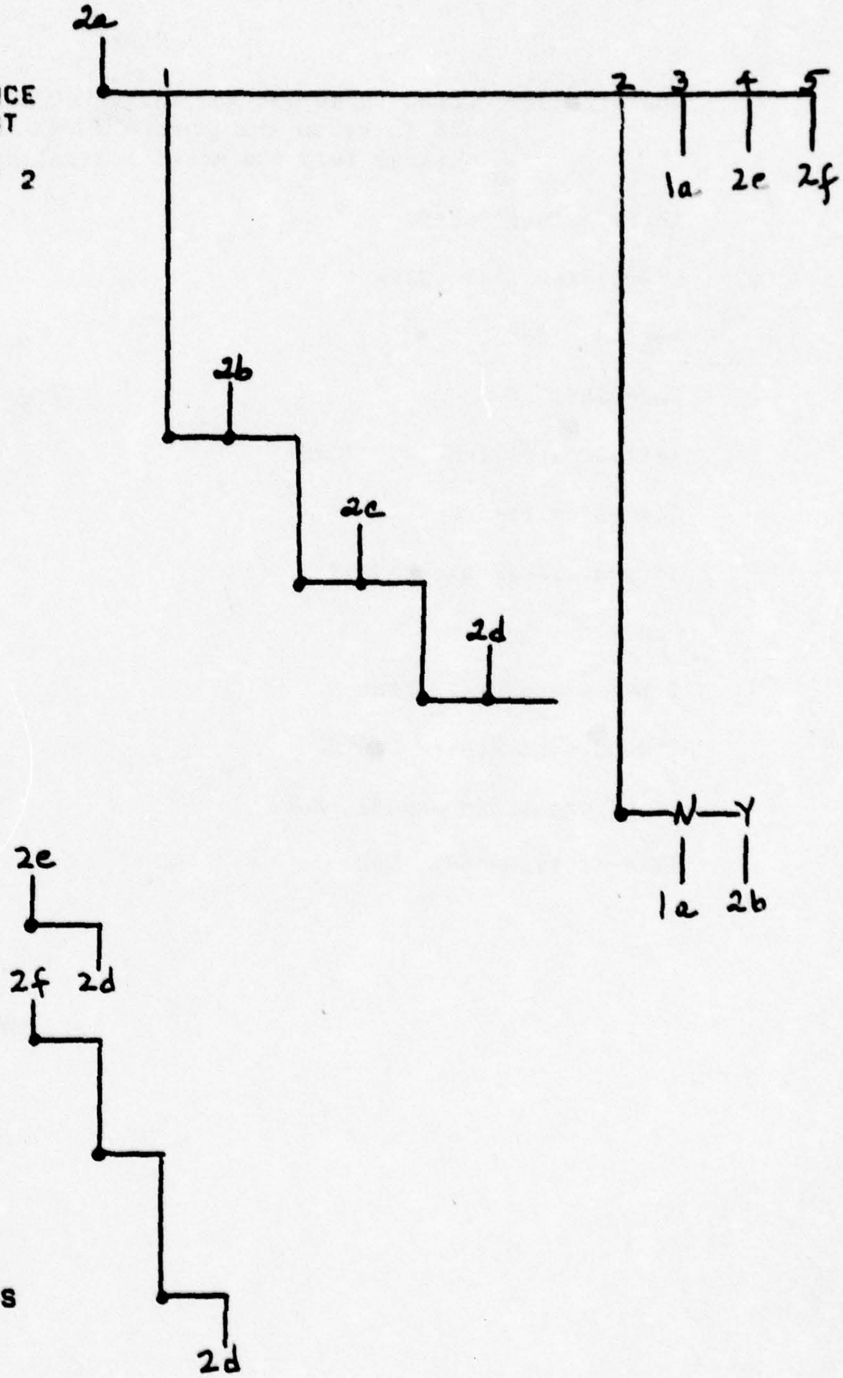
IS RANGE <= 2.5 MILES?

START WAVE IN 10 SECONDS

WVTM = 15 SECONDS

IF RANGE < = 1.5 MILES,
WVTM = 8 SECONDS

START WAVE IN WVTM SECONDS



TOWER FLOWCHART (SHEET 2 OF 2)

CLRNC

Description: CLRNC turns out the white REQUEST light on the student's panel and turns on the green CLEARED light. It also sets the wind message into the model controller queue.

Entry point: CLRNC

Classification: Task

Period: None

Language: F

Activated/called by: TOWER

Cancelled by: CLRNC

Activates/calls: PANOUT

Cancels: None

Input arguments: None

Common variables: CTOTHR

Files created/changed: None

Files referenced: None

```
PROCEDURE                         CLRNC

CALL PANOUT TO TURN OUT
REQUEST LIGHT

SET CTOTHR TO WIND MESSAGE

CALL PANOUT TO TURN ON THE
CLEARED LIGHT

DIE
```

CLRNC FLOWCHART (SHEET 1 OF 1)

IGNORE

Description: IGNORE turns out the REQUEST light on the trainee panel.

Entry point: IGNORE

Classification: Task

Period: None

Language: F

Activated/called by: TOWER

Cancelled by: IGNORE

Activates/calls: PANOUT

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE **IGNORE**

CALL PANOUT TO TURN OUT
CLEARANCE REQUEST LIGHT

DIE

IGNORE FLOWCHART (SHEET 1 OF 1)

WAVE

Description: WAVE conveys the waveoff order to the final controller. It also sets the waveoff message into the set of acceptable SUS messages.

Entry point: WAVE

Classification: Task

Period: None

Language: F

Activated/called by: TOWER

Cancelled by: WAVE

Activates/calls: PANOUT

Cancels: None

Input arguments: None

Common variables: CTOTHR, PTAPR

Files created/changed: None

Files referenced: None

PROCEDURE

CALL PANOUT TO TURN OFF
REQUEST

CALL PANOUT TO TURN OFF
CLEARED

BRANCH ON APPROACH TYPE
(PTAPR)
1-FULL STOP
2-LOW APPROACH
3-TOUCH AND GO

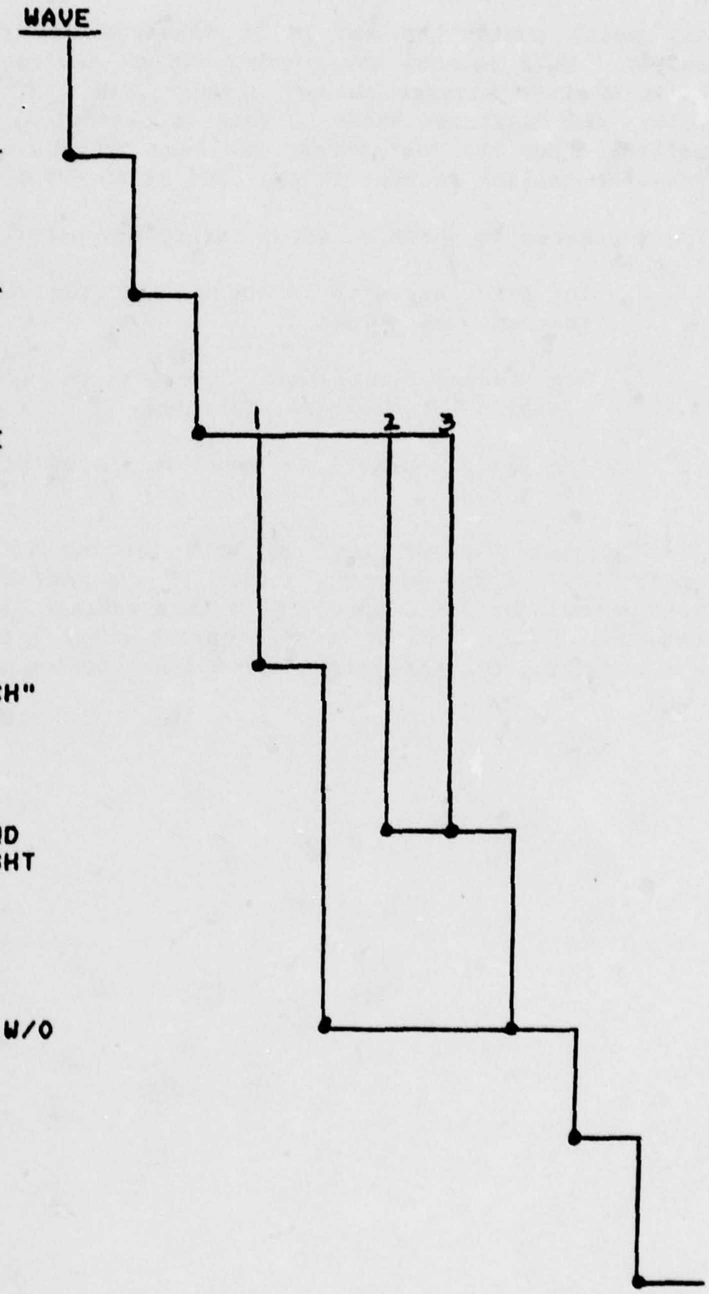
SET CTEMERG TO
"EXECUTE MISSED APPROACH"

SET CTOTHR TO "CLIMB AND
MAINTAIN 1500, TURN RIGHT
HEADING 300"

CALL PANOUT TO TURN ON W/O

CALL PANOUT TO TURN ON
AUDIBLE ALARM

DIE



WAVE FLOWCHART (SHEET 1 OF 1)

Automated Voice (GLIB)

All model controller and pilot messages are channeled through GLIB for output. GLIB selects the correct output device for the message and determines whether message output is appropriate at the time. Overlapping of Votrax and digitized audio outputs is prevented. A notice is placed in the mailbox after the last phrase has been output. If the message is not output, the calling routine is notified by an error return value of minus one.

The instances in which an error return may occur follow:

1. The pilot attempts to speak while the mike is keyed and the pilot's frequency is selected.

2. The feeder controller attempts to speak when the correct ICS position button is not selected.

3. The pilot attempts to speak while the proper frequency is not being monitored.

Another function performed by this routine is the verification of final controller message output status. If a phrase cited in the message has not been authorized for output, the entire message is not output and no error is returned. This feature is used during phase 1 training demonstrations when a restricted vocabulary is desired for teaching purposes.

GLIB

Description: GLIB qualifies and schedules all message output from the final controller, pattern controller, and pilot models. The messages are queued for output to avoid message overlaps.

Entry point: GLIB

Classification: Subroutine

Period: None

Language: F

Activated/called by: PICKY, FEED, PILOT

Cancelled by: N/A

Activates/calls: Digitized voice and Votrax controllers, DONE

Cancels: None

Input arguments: IARG1 — message source (1 = final, 2 = feeder, 3 = pilot)
IARG2,.....IARGN — Phrase identifiers

Common variables: PRATL, PRDEV, PRQUE, PRDNE, PRENT, PRSAY, BXGLB, PRES, PRAC, CTON

Files created/changed: Activity replay file

Files referenced: None

BRANCH ON MESSAGE SOURCE

HAVE ALL THE PHRASES IN THE MESSAGE BEEN AUTHORIZED FOR OUTPUT?

FIND THE OUTPUT DEVICE

IS THE PROPER ICS BUTTON SELECTED?

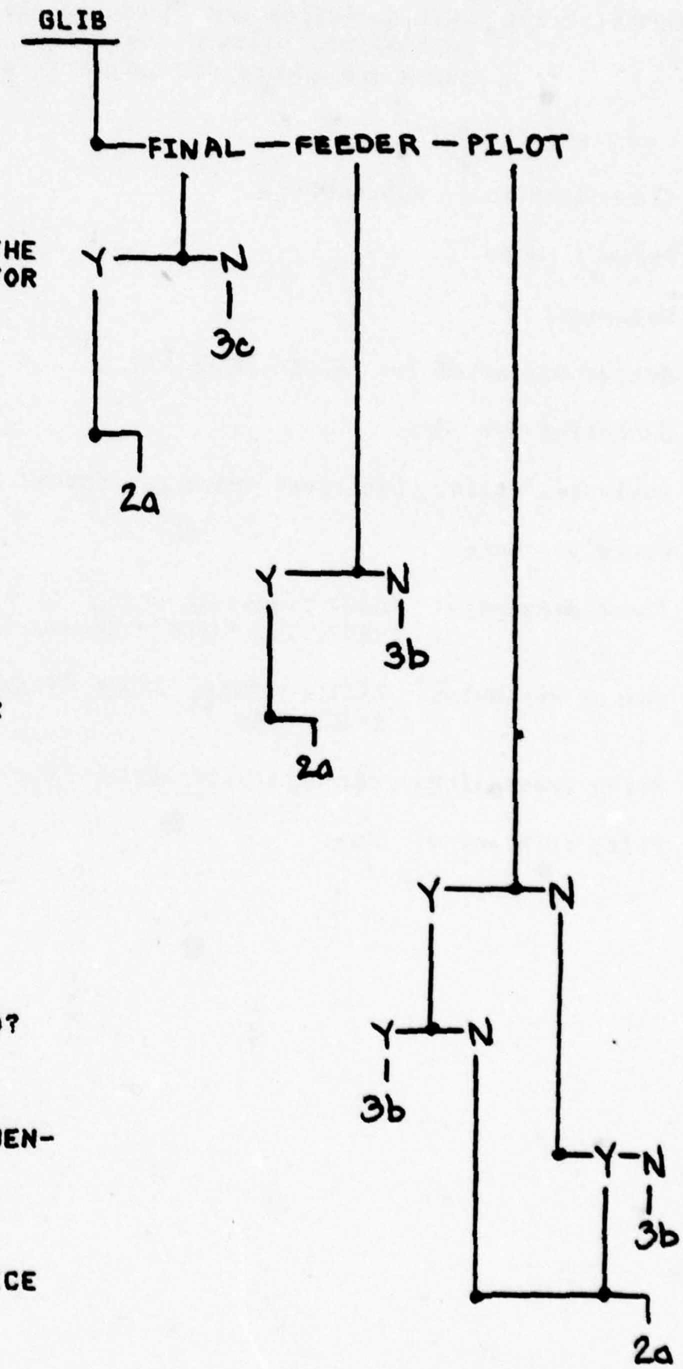
USE THE FEEDER CONTROLLER OUTPUT DEVICE

IS THE PROPER RADIO FREQUENCY BUTTON SELECTED?

IS THE FOOT KEY DEPRESSED?

IS THE PROPER RADIO FREQUENCY BEING MONITORED?

USE THE PILOT OUTPUT DEVICE



GLIB FLOWCHART (SHEET 1 OF 3)

IS THERE A MESSAGE WAITING IN THE QUEUE?

PUT THE OUTPUT DEVICE AND MESSAGE PHRASES INTO THE QUEUE

RECORD MESSAGE IDENTITY

IS THE CRT THE OUTPUT DEVICE?

SET THE DONE FLAG FALSE

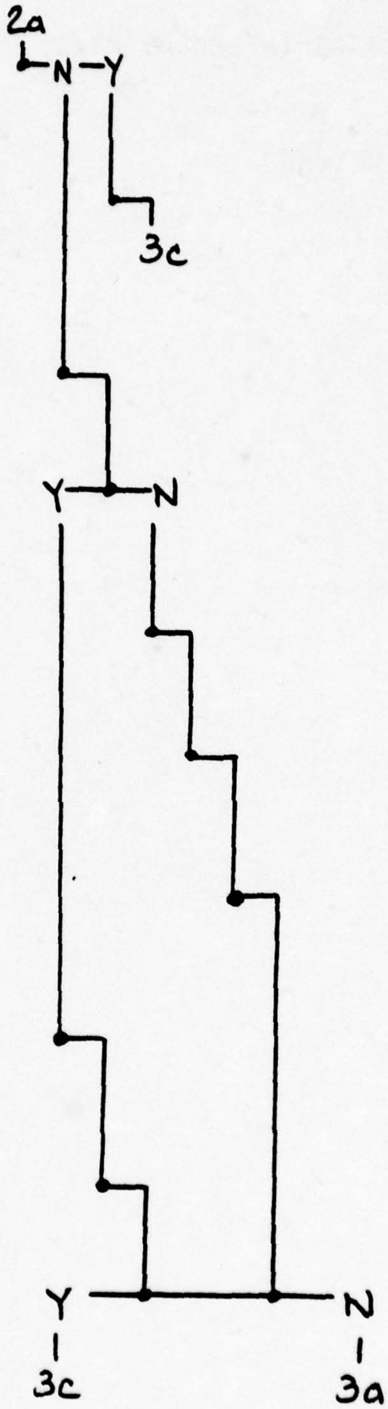
SEND MESSAGE OUTPUT REQUEST TO DEVICE CONTROLLER

START TASK WHICH WAITS FOR THE MESSAGE OUTPUT

SEND MESSAGE REQUEST TO CRT CONTROLLER

SEND OUT MESSAGE IDENTITY

IS THIS A DEMO?

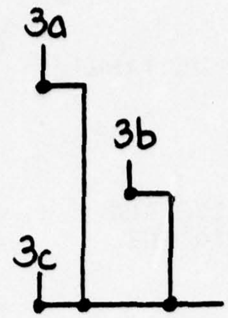


GLIB FLOWCHART (SHEET 2 OF 3)

RECORD MESSAGE IN REPLAY FILE

SET ERROR RETURN

RETURN



GLIB FLOWCHART (SHEET 3 OF 3)

DONE

Description: This task awaits the end of Votrax or audio output. If a message is in the output queue, it is sent to the appropriate device controller.

Entry point: DONE

Classification: Task

Period: None

Language: F

Activated/called by: GLIB

Cancelled by: None

Activates/calls: Digitized voice and Votrax controllers.

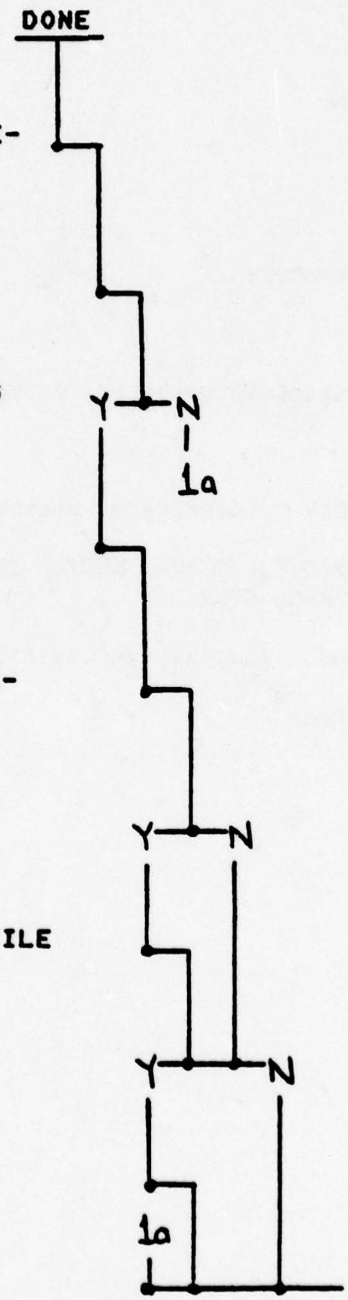
Cancels: None

Input arguments: IDEV — identity of speech device in use

Common variables: PRQUE, PRDNE, PRENT, PRSAY, BXGLB, PRES, EVSPT, EVVRO, PRAC, CTON

Files created/changed: Activity replay file

Files referenced: None



WAIT FOR THE END OF THE DE-
VICE OUTPUT

SEND OUT MESSAGE IDENTITY

IS THERE A MESSAGE WAITING
IN THE QUEUE?

RECORD MESSAGE AND DEVICE
IDENTITY

SEND MESSAGE REQUEST TO DE-
VICE CONTROLLER

IS THIS A DEMO?

RECORD MESSAGE IN REPLAY FILE

IS THE OUTPUT VIA CRT?

SET DONE FLAG

DIE

DONE FLOWCHART (SHEET 1 OF 1)

Performance Measurement

This section describes the set of criterion referenced indicators which will be used to meet GCA-CTS performance requirements, as well as the routines which will collect and format the data.

Performance Measurement Variables

Performance data are required to provide the following capabilities:

1. Real-time error detection and freeze in phase 2.

2. Adaptive problem selection.

3. Student feedback in the form of annotated replay and performance summaries.

4. Instructor feedback emphasizing overall progress.

5. Performance test scoring after misrecognition correction.

For some purposes, such as for freezing in phase 2, only an error indicator is needed. Adaptive problem selection, on the other hand, requires a quantitative measure of performance on specific tasks in which errors are weighted by their relative importance. In the performance measurement variable (PMV) descriptions in the tables which follow, both the error indicators and the contributions to the particular PMV score are defined. All PMV scores are in the range $0 \leq s \leq 100$.

TABLE 18. PV01, ACCEPT HANDOFF COMPOSITE

	Bit of PV01	Partial Credit	Total Possible Points
A. Monitor feeder controller ICS	1		10
B. Monitor proper frequency as specified in the handoff	2		10
C. Acknowledge handoff			20
1) Acknowledgement given prior to radar contact	3	10	
2) Acknowedgement given within 10 seconds	4	10	
D. Report radar contact			50
1) Radar contact reported prior to radio check	5	10	
2) 50% of target on display at report	6	15	
3) Report not later than 10 seconds after 50% target appearance	7	15	
4) Call sign correct	8	5	
5) Radio frequency correct	9	5	
E. ICS off, radio frequency selected			10
1) If pattern does not relinquish frequency, "give me..." request is made within 15 seconds; otherwise not.	10,11	5	
2) When pattern relinquishes frequency, ICS is deselected.	12	5	

TABLE 19. PV02, RADIO CHECK COMPOSITE

	Bit of PV02	Partial Credit	Total Possible Points
A. Radio Contact			70
1) Within 30 seconds of 50% target appearance	1	10	
2) Proper frequency selected	2	10	
3) Mike keyed	3	10	
4) Call sign used	4	10	
5) One of the following given: a) "How do you hear..." b) "Wheels..." c) "Turn...heading" d) "Turn..."	5	10	
6) Unkey within 3 seconds and remain unkeyed 5 seconds	6	20	
B. Speech quality			30
1) Pilot responds "Loud and clear,"	7	30	
or			
2) If pilot responds "Weak...,"			
a) Student answers "how...now," unkeys within 3 seconds and remains unkeyed 5 seconds	8	15	
b) Pilot can respond "Loud...," i.e., V.U. level normal	9	15	

TABLE 20. PV03, TURN TO FINAL COMPOSITE

	Bit of PV03	Partial Credit Turn	Partial Credit Straight In	Total Points Possible
A. Accuracy of turn advisories, if given. (Score is given a weight of .6, score for B weighted .4; for a straight-in approach, the entire 100 points is given on B 1 and 2)				60
1) Number of turns to reach a heading within 10° of runway heading (maximum of 3 turns scored)	13-15			
2) Turn in proper direction	7,10,13	25		
3) Allowance for wind	8,11,14	25		
4) Call sign correct	9,12,15	10		
B. Quality of turn or initial control				40-100
1) At 6 miles (3 for short approach) target is within 2 target widths of cursor	1	10	30	
2) At 5 miles (2 for short approach) target must intercept azimuth cursor in target zone 1 or 2	2	20	70	
3) More than 1 turn should be used to turn aircraft into final	3	10		

TABLE 21. PV04, APPROACHING GLIDEPATH COMPOSITE

	Bit of PV04	Partial Credit	Total Possible Points
A. Approaching glidepath			25
1) Advisory given	0	10	
2) Correct call sign and over used, if needed; otherwise not.	1,2	5	
3) Advisory given when aircraft is within the correct range	3	5	

Aircraft Speed	Acceptable Range (Miles)
90	0.25-0.75
120	0.33-1.00
140	0.38-1.16
160	0.44-1.33
200	0.55-1.67

	Bit of PV04	Partial Credit	Total Possible Points
4) Advisory transmitted only once during final approach	4	5	
B. Do not acknowledge			25
1) Advisory given	5	10	
2) Correct call sign used	6	5	
3) The phrase is not followed by "over"	7	5	
4) Transmitted prior to begin descent	8	5	
C. Begin			30
1) Advisory given	9	10	
2) Transmitted within 10-30 seconds after approaching glidepath	10	5	
3) Glidepath cursor intersects upper 1/3 of target	11	10	
4) Transmitted only once during the approach	12	5	
D. Wheel check			20
1) Advisory must be given unless pilot responds "wheels down" to radio check, in which case it must not be given	13,14	15	
2) Correct call sign and over must	15	5	

TABLE 22. PV05, HEADING ADVISORIES COMPOSITE

	Bit of PV05(0)	Array Element of PV05 Error	Array Element of PV05 Counter	Weighting Factor Applied to Percentage Error
A. Range greater than 5 miles; all turns must be evenly divisible by 5°	1	1	9	.1
B. Range less than or equal to 5 miles, turns must not be of 1°	2	2	10	.1
C. All heading advisories				
1) Direction of the turn and heading digits must correspond such that the direction advised causes the smaller turn	3	3	11	.2
2) A counter-corrective turn must be made within 8 seconds when a turn of more than 120° is given	4	4	12	.05
3) If the target enters zone 3 from zone 2, a heading correction must be given within 20 seconds. This check is initiated when target has been in zones 1 or 2 for 1/2 mile, or at 5 miles (2 for short approach), whichever comes first.	5	5	13	.15
4) Assigned headings must include compensation for wind (crab)	6	6	11	.1
5) The heading given in the "Heading..." message must be the same as previously assigned	7	7	14	.2
6) "Heading..." must not be used more than 5 times in an approach	8	8	14	.1

TABLE 23. PV06, AZIMUTH POSITION AND TREND COMPOSITE

	Bit of PV06(0)	Array Element of PV06 Error	Array Element of PV06 Counter	Weighting Factor Applied to Percentage Error
A. Position calls				
1) Position call correct	1	1	5	.5
2) "Well" followed by a corrective turn within 3 seconds, or "correcting"	2,3	2	6	.25
B. Trend calls				
"Correcting" must be used only when target is closing with centerline	4	4	7	.25

TABLE 24. PV07, GLIDEPATH POSITION AND TREND COMPOSITE

	Bit of PV07(0)	Array Element of PV07 Error	Counter	Weighting Factor Applied to Percentage Error
A. For all glidepath messages, begin descent must have been given	1	1	8	.10
B. Position calls				
1) Position must be correct	2	2	9	.15
2) A position call must be made whenever target changes zones, unless superseded by a priority call	3	3	10	.15
C. Trend Calls				
1) Trend must be correct	4	4	11	.15
2) Trend must be issued if the target moves from one zone to another	5	5	11	.15
3) Trends must not be issued successively except in well zone	6	6	11	.15
4) Trends must not separate identical position messages except in well zone	7	7	11	.15

TABLE 25. PV08, RANGE CALL COMPOSITE

	Bit of PV08(0)	Array Element of PV08 Error	Counter	Weighting Factor Applied to Percentage Error
A. All range calls must be made once the first one is made or 5 miles is reached, whichever comes first, unless superseded	11	1	5	.6
B. The call must be made within ± 0.1 mile of the mark	12	2		.1
C. Correct miles must be used	13	3		.2
D. Call must not be repeated	14	4		.1

TABLE 26. PV09, DECISION HEIGHT COMPOSITE

	Bit of PV09	Partial Credit	Total Possible Points
A. Decision height call			60
1) If target is touching both cursors, ADH given only	1		
or			
2) If target is not touching,			
a) ADH given	1	30	
b) followed by highest priority correct position	2,3,4,5,6	30	
B. Range			30
1) DH announced within .85 miles from touchdown	7	10	
2) DH announced prior to .65 miles from touchdown	8	20	
C. Call is made only once during the approach	9		10

TABLE 27. PV10, CLEARANCE COMPOSITE

	Bit of PV10	Partial Credit	Total Possible Points
A. Clearance request	14,15		50
1) Initial clearance request is made after 3.1 miles	1	10	
2) Initial clearance request is made prior to or at 2.9 miles	2	30	
3) If clearance is not received, a second request is posted between 2.1 and 1.9 miles; else no further request	3,4	10	
B. Issuance of clearance when received received from tower			50
1) Correct wind information is given	5	10	
2) Wind is issued after clearance is received from tower	6	10	
3) Clearance is issued after wind	7,8	10	
4) Clearance must be issued prior to 1 mile	9	20	
or			
C. Clearance problems leading to a waveoff			
1) If clearance is not received			
a) Issue waveoff prior to 1.3 miles	10	35	
b) Use proper missed approach advisory	11	15	
or			
2) If waveoff is given or clearance is cancelled			
a) Issue waveoff within 2 seconds of receipt of cancellation	12	35	
b) Use proper missed approach advisory	11	15	

TABLE 28. PV11, OVER LANDING THRESHOLD COMPOSITE

	Bit of PV11	Partial Credit	Total Possible Points
A. Over landing threshold			40
1) Advisory given	1	20	
2) Given within ±1 second of the target contacting the landing threshold point	2	20	
B. Final course position			60
1) Given within 3 seconds of OLT	3	20	
2) Position correct (including "over" for on position)	4	20	
3) Over is used correctly	5	20	

TABLE 29. PV12, HANDOFF AND ROLLOUT COMPOSITE

	Bit of PV12	Partial Credit	Total Possible Points
			100
A. Rollout instructions on full stop landing			
1) Rollout instructions given	1	40	
2) Instructions issued 20-40 seconds after "over"	2	20	
3) Release radio frequency within 10 seconds after rollout instructions	3	20	
4) Notify pattern controller	4	20	
or			
B. Handoff to the pattern controller made if aircraft is on low approach or touch and go, or executing a missed approach including lost communications			
1) Handoff is given	5	40	
2) Handoff is made within 30 seconds of:	6	10	

Condition	Reference Point
Waveoff	Issuance of waveoff
Low approach	Decision height
Touch and go	Landing threshold

	Bit of PV12	Partial Credit	Total Possible Points
3) Call sign and buttons correct	7,8,9	10	
4) If missed approach, range must be given to nearest 1/2 mile, else not	10,11	10	
5) Monitor frequency and ICS until pattern transmits "CS radar"	12	10	
6) Release radio frequency	13	10	
7) Pattern ICS selected during handoff	14	10	

TABLE 30. PV13, NO-GYRO COMPOSITE

	Bit of PV13	Partial Credit	Total Possible Points
A. Warn pilot			20
Give "heading XXX" if 1/4 mile elapses after a turn and less than a 2° change in course is observed			
B. Prepare for no-gyro			50
1) Announce no-gyro approach	1	30	
2) Announce no-gyro approach if course correction is not taken within 1/2 mile	2	10	
3) Issue the announcement prior to 3/4 mile from the point at which warning was issued	3	10	
C. Make 1/2 standard rate turns			30
1) Give advisory	4	10	
2) Issued after begin descent, and no-gyro announcement	5	10	
3) Transmitted only once	6	10	

TABLE 31. PV14, NO-GYRO HEADING COMPOSITE

	Array Element of PV14 Error	Array Element of PV14 Counter	Weighting Factor Applied to Percentage Error
A. Turn must be in correct direction	1	4	.4
B. Stop turn must be issued	2		.4
C. If target enters zone 3 from zone 2, a heading correction must be given within 20 seconds	3	5	.2

TABLE 32. PV15, EMERGENCY WAVEOFFS

	Bit of PV15	Partial Credit	Total Possible Points
			100
A. Radar contact lost			
1) If target moves off the display or the display fails, issue waveoff	1	50	
2) Issued within 5 seconds	5	25	
3) Proper R/T for type of approach	6	25	
or			
B. Minimum separation			
1) If targets get within 2.8 miles of one another, issue waveoff	2	50	
2) Issued within 5 seconds	5	25	
3) Proper R/T for type of approach	6	25	
or			
C. Erratic maneuvers			
1) If target moves from one well zone to another in 3 seconds, issue waveoff	3	50	
2) Issued within 10 seconds	5	25	
3) Proper R/T for type of approach	6	25	
or			
D. Target not touching at decision height			
1) Target not touching when decision height message given and waveoff issued	4	50	
2) Follows "Too..." message immediately	5	25	
3) Proper R/T for type of approach	6	25	

TABLE 33. PV16, LOW ALTITUDE ALERT

	Bit of PV16	Partial Credit	Total Possible Points
A. Low altitude alert			100
1) Transmitted when target exceeds 1 target width per mile below glidepath	1	50	
2) Issue within 5 seconds	2	50	

TABLE 34. PV17, TRANSMISSION BREAK

	Bit of PV17(0)	Array Element of PV17		Weighting Factor Applied to Percentage Error
		Error	Counter	
A. Mike unkeyed after "over"	1	1	3	.8
B. At least one given subsequent to do not acknowledge prior to 1 mile	2	2	4	.2

TABLE 35. PV18, TRANSMISSION RATE

	Bit of PV18(0)	Array Element of PV18		Weighting Factor Applied to Percentage Error
		Error	Counter	
A. Transmission rate after do not acknowledge advisory				
1) At least 1 second between advisories	1	1	3	.5
2) Not more than 5 seconds between advisories	2	2	3	.5

TABLE 36. PV19, RADAR ALIGNMENT COMPOSITE

	Bit of PV19	Partial Credit	Total Possible Points
A. Alignment check preparation			20
1) Azimuth: servo down until centerline reflector appears	7	10	
2) Elevation and range: servo left until touchdown reflector appears	8	10	
B. Select align if alignment of			60
1) Azimuth	1,2	20	
2) Elevation	3,4	20	
or			
3) Range	5,6	20	
is needed; else not			
C. Reposition antennae			20
1) Servo up until 1 mile mark is bisected by glideslope	9	10	
2) Servo right until the 1 mile mark is bisected by azimuth cursor	10	10	

A, B, and C must be performed sequentially, nor no credit is given.

Performance Measurement Routines

One set of performance measurement routines was designed to satisfy the diverse GCA-CTS requirements. These routines operate independently of one another, in real time during phase 2, and after the run during replay of phase 3. One simple interface routine makes this possible. It merely reads data formatted as though for output to the replay activity file in phase 2, or from the activity file input buffer during replay. In general, after initialization, a performance measurement executive routine is responsible for activating particular PMV modules when events occur. Program descriptions are provided for these routines with the exception of the enrichment PMVs. String charts are also provided for most routines. In these program descriptions, the actual bit numbers, etc., are used, making cross reference to the PMV descriptions easy. Of course, the actual code will use mnemonic parameters.

PMS

Description: PMS causes the PMVs to be scored.

Entry point: PMS

Classification: Task

Period: None

Language: F

Activated/called by: P23SUB, REPLAY

Cancelled by: PMS, RDRPLY

Activates/calls: HRPMS, P***, EXPLAIN, RDACT, TIMSCHD

Cancels: None

Input arguments: None

Common variables: PV**, KY***

Files created/changed: None

Files referenced: None

PROCEDURE

WAS A HANDOFF GIVEN?

CALL HRPMS TO SCORE HANDOFF PROCEDURE

UPDATE RANGE TO CORRESPOND TO LAST INPUT

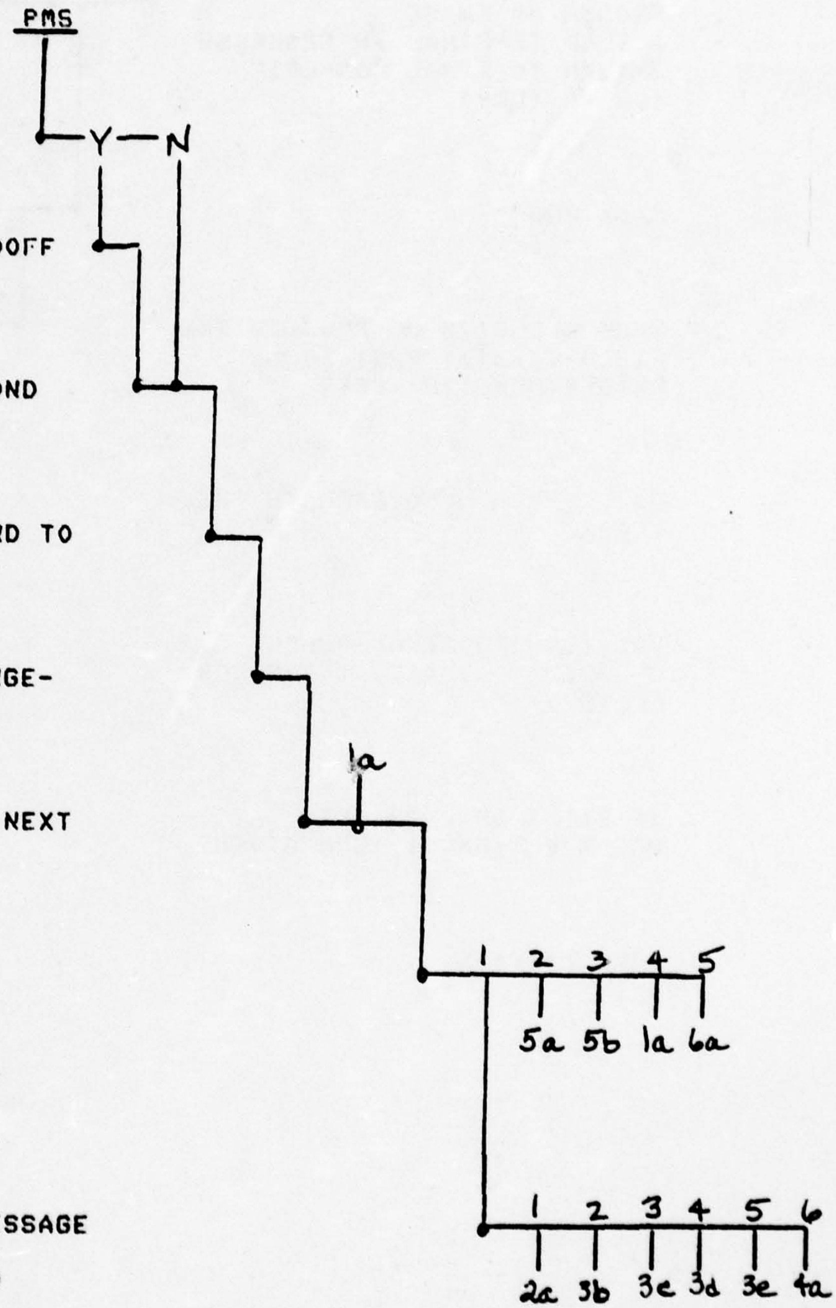
UPDATE TIME TO CORRESPOND TO TIME OF LAST INPUT

CALL TO SCHEDULE ANY RANGE-RELATED

CALL RDACT (-1) TO READ NEXT ACTIVITY

BRANCH ON NEXT ACTIVITY
1-MODEL CONTROLLER /SUS
2-PANEL CHANGES
3-SERVO CHANGES
4-AUTOMATED VOICE OUTPUT
5-SPECIAL RECORDS

BRANCH ON TYPE OF SUS MESSAGE
1-TURNS
2-COURSE POSITION, TREND
3-GLIDEPATH POSITION
4-GLIDEPATH TREND
5-MILE MARKS
6-OTHER



PMS FLOWCHART (SHEET 1 OF 6)

BRANCH ON PHASE
1-TURN TO FINAL IN PROGRESS
2-TURN TO FINAL COMPLETE
(OR NO TURN)

CALL P03A

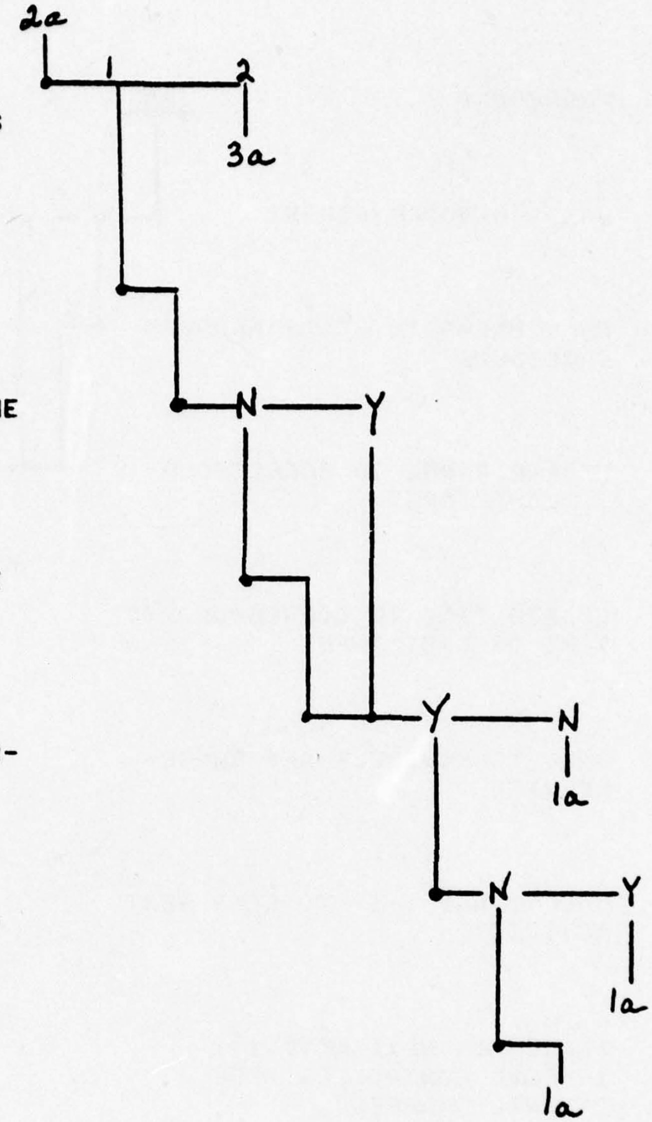
WERE ALL BITS OF PV03 IN THE
FIELD (4+3*#TURNS) TO
(4+3*#TURNS)+2) SET?

CALL EXPLAIN TO EXPLAIN THE
ERROR

WAS TURN TO FINAL PHASE
CHANGED TO 2 (IE. TURNS COM-
PLETE?)

IS BIT 3 OF PV03 SET, IE.
WAS MORE THAN 1 TURN GIVEN?

CALL EXPLAIN



PMS FLOWCHART (SHEET 2 OF 6)

3a

BRANCH ON TYPE OF FLIGHT 1 2
1-NORMAL
2-NOGYRO

CALL P05 TO SCORE TURNS ETC.

1a

CALL P14 TO SCORE NO GYRO
TURNS ETC.

1a

3b

CALL P06 TO SCORE POSITION
CALLS

1a

3c

CALL P07A TO SCORE GLIDEPATH
POSITION

1a

3d

CALL P07B TO SCORE GLIDEPATH
TREND

1a

3e

CALL P08 TO SCORE MILE MARKS

1a

PMS FLOWCHART (SHEET 3 OF 6)

4a

CALL SPECIFIED ROUTINE BASED
ON TYPE OF OTHER MESSAGE

1a

1-APPROACHING GLIDEPATH:
P04A
2-DO NOT ACKNOWLEDGE:
P04B
3-BEGIN DESCENT:
P04C
4-WHEEL CHECK:
P04D
5-WIND:
P10A
6-CLEARANCE:
P10B
7-DECISION HEIGHT:
P09A
8-TOO FAR:
P09B
9-LANDING THRESHOLD:
P11
10-HANDOFF:
P12A
11-ROLLOUT:
P12B
12-RADAR LOST:
P15
13-WAVE OFF:
P15
14-LOW ALTITUDE ALERT:
P16
15-OVER:
P17A STARTED IN 3 SECONDS

PMS FLOWCHART (SHEET 4 OF 6)

PANEL CHANGE, SET KY** VARIABLE

WAS IT MIKE UNKEY AFTER 5 MILES?

INCREMENT P17(4)

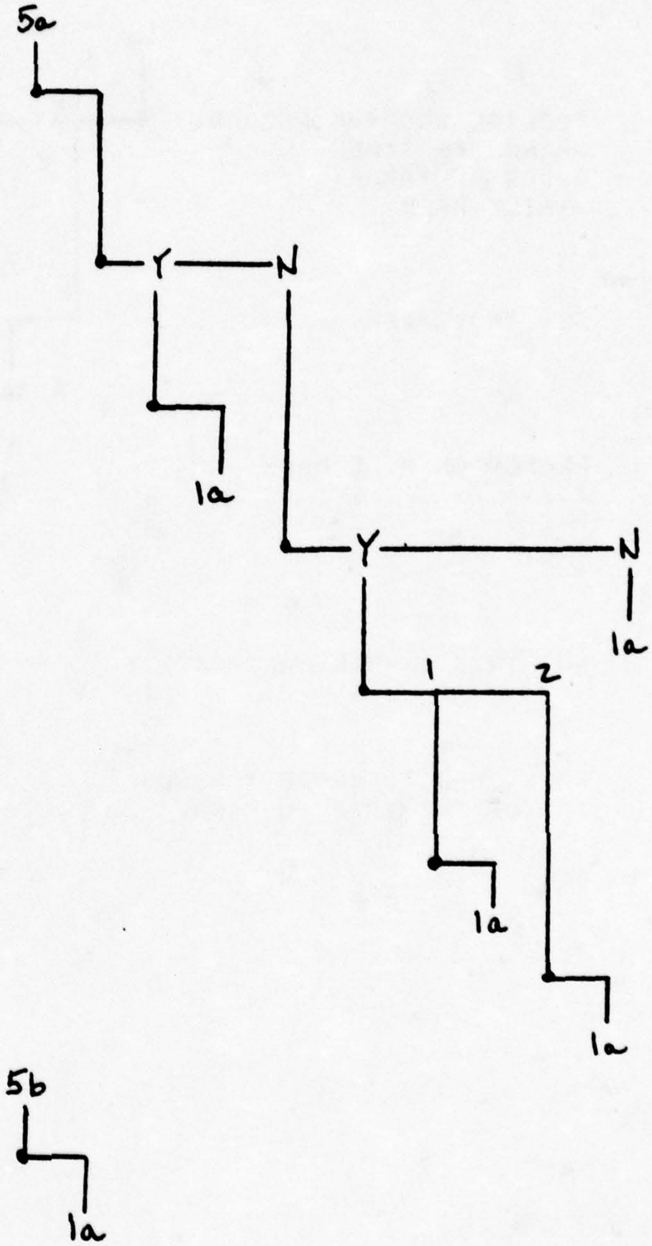
WAS IT STUDENT VOICE ACTIVE CHANGE?

BRANCH ON ACTIVITY
1-START OF VOICE
2-END OF VOICE

CALL P18A

SCHEDULE P18B IN 5 SECONDS (TIMSCHD)

SERVO CHANGE, NOTE NEW POSITION



PMS FLOWCHART (SHEET 5 OF 6)

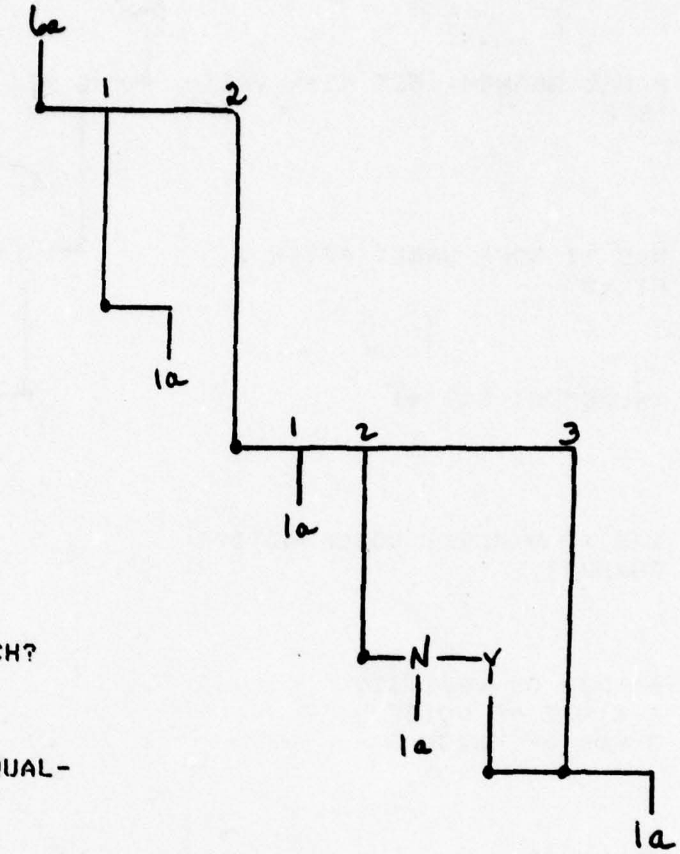
SPECIAL SCORING RECORD.
BRANCH ON TYPE
1-50% OF TARGET
2-MILE MARK

SET INDICATOR

BRANCH ON MILE MARK
1-1,4,7+
2-2,3
3-5,6

WAS THIS A SHORT APPROACH?

CALL P03B TO SCORE THE QUAL-
ITY OF THE TURN TO FINAL



PMINT

Description: This routine queues performance event detector routines depending upon whether or not the PMV is scored. It also initializes PMV flags and indicators.

Entry point: PMINT

Classification: Subroutine

Period: None

Language: F

Activated/called by: P23SUB

Cancelled by: N/A

Activates/calls: PM2INT, PM3INT, PMDINT

Cancels: None

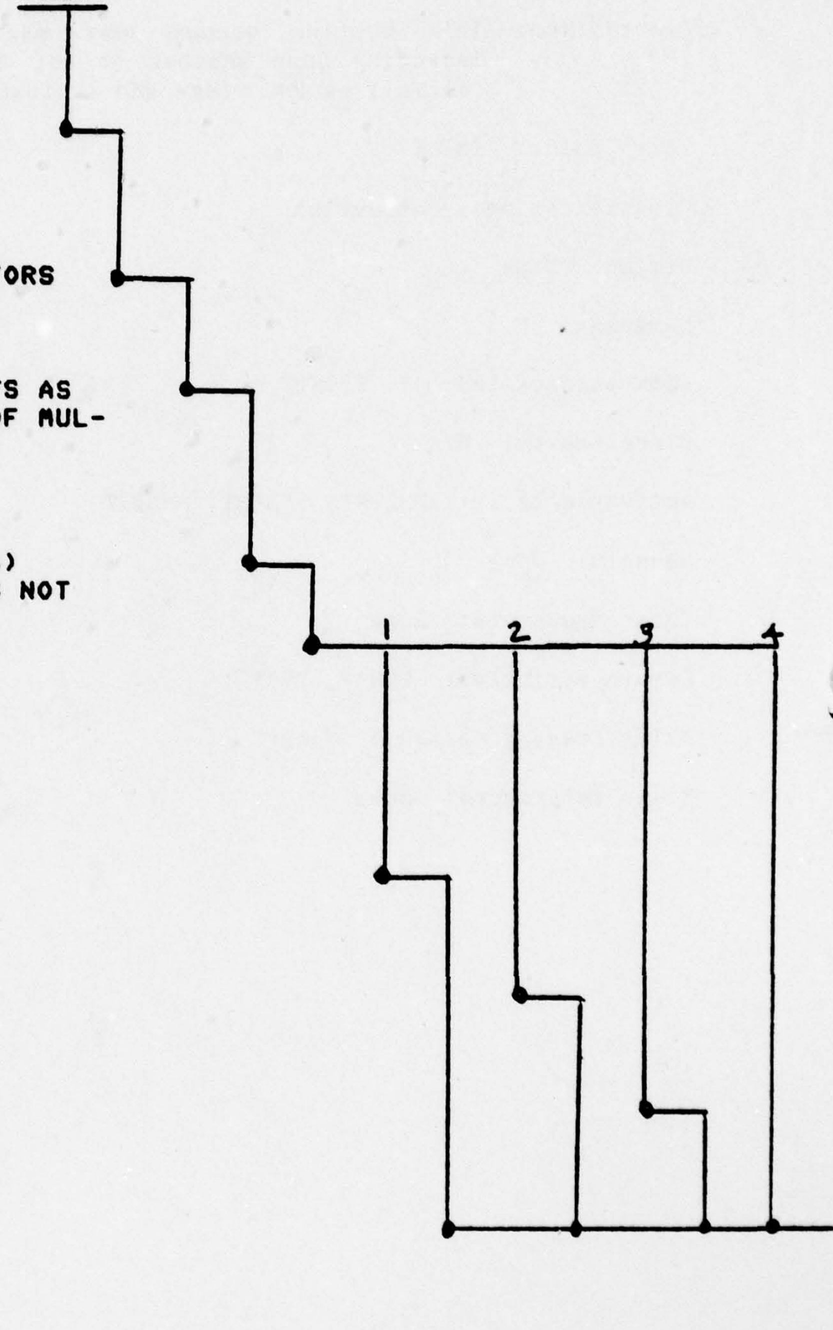
Input arguments: None

Common variables: PVN**, PV**

Files created/changed: None

Files referenced: None

PROCEDURE PMINT



INITIALIZE CONTROLLER MESSAGES

INITIALIZE PMS INDICATORS

SET ADVISORY GIVEN BITS AS NEEDED FOR DETECTION OF MULTIPLE ADVISORIES

SET BITS 1-5 OF PV08(6) TO INDICATE MILE MARKS NOT GIVEN
BRANCH ON PHASE
1-PHASE 2
2-PHASE 3, P-RUN
3-DEMO
4-REPLAY

START PM2INT

START PM3INT

START PMDINT

RETURN

PMINT FLOWCHART (SHEET 1 OF 1)

PM2INT

Description: This routine sets up the range-related model controller advisories and schedules omission checks in phase 2 problems.

Entry point: PM2INT

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMINT

Cancelled by: N/A

Activates/calls: TGT50, PCHK, RNGSCHD

Cancels: None

Input arguments: None

Common variables: PFS**

Files created/changed: None

Files referenced: None

PROCEDURE PM2INT

FREEZE ON ERROR IN HANDOFF?

START TGT50

FREEZE ON TURN TO FINAL
ERROR?

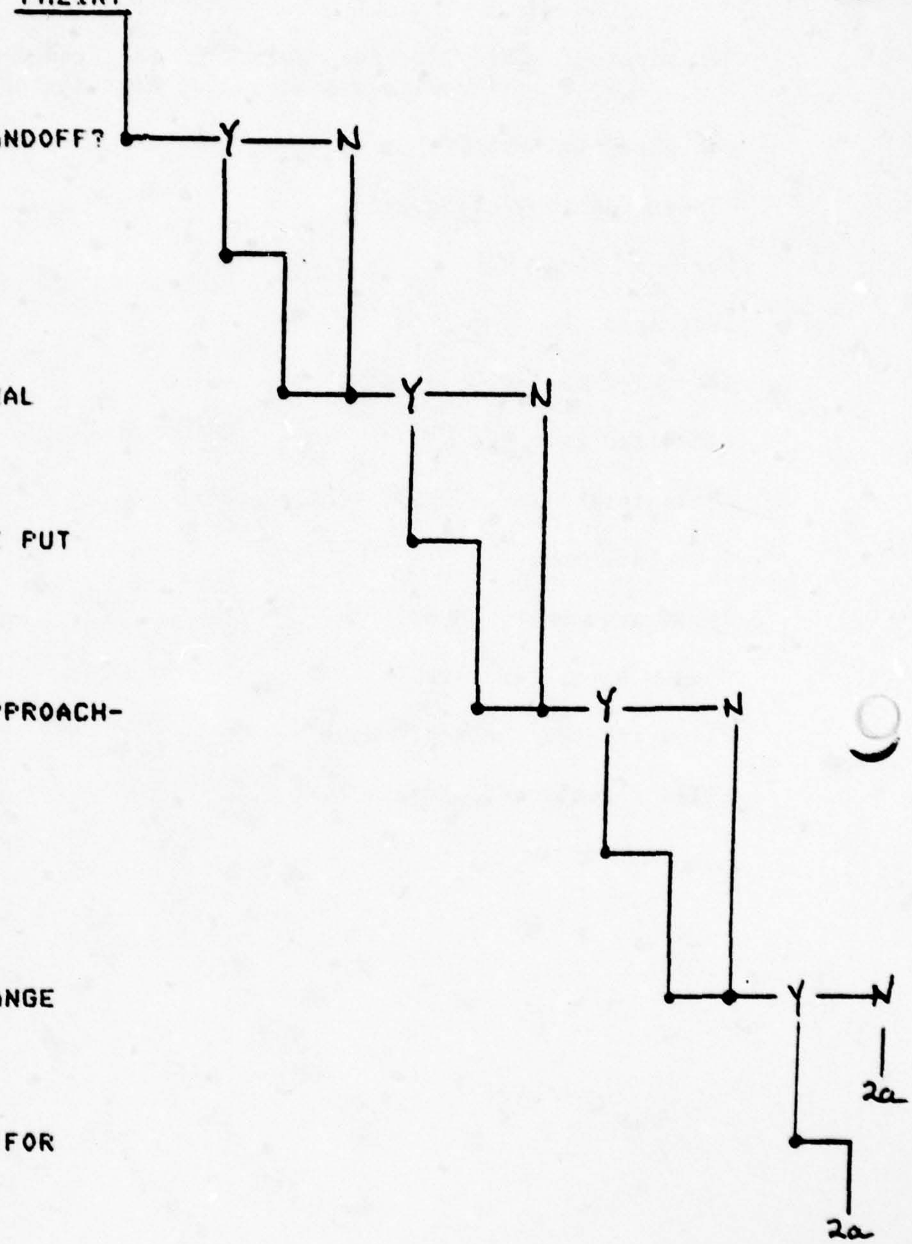
CALL PCHK TO SCHEDULE PUT
AND CKTTF

FREEZE ON ERROR IN APPROACH-
ING GLIDEPATH?

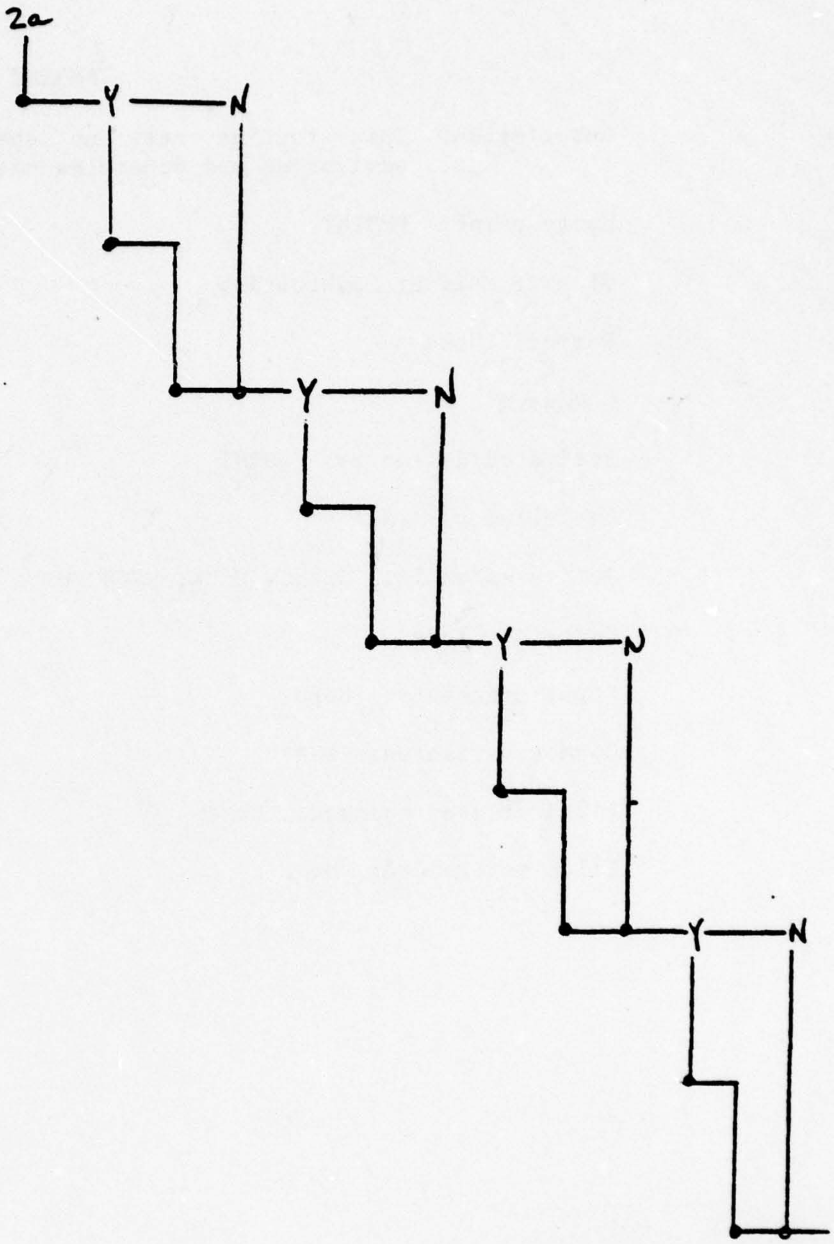
CALL PCHK (CKAGP...)

FREEZE ON ERROR IN RANGE
CALLS?

CALL PCHK (CKRNG...) FOR
5-1 MILES



PM2INT FLOWCHART (SHEET 1 OF 2)



FREEZE ON DECISION HEIGHT
ERROR?

CALL PCHK (CKADH...)

FREEZE ON CLEARANCE ERROR?

CALL RNGSCHD

FREEZE ON OVER LANDING
THRESHOLD ERROR?

CALL PCHK (CKOLT...)

FREEZE ON HANDOFF OR
ROLLOUT?

CALL PCHK (CKHO...) AFTER
LOW APPROACH, ETC.

RETURN

PM2INT FLOWCHART (SHEET 2 OF 2)

PM3INT

Description: This routine sets up the range-related model controller advisories and schedules omission checks for phase 3 problems.

Entry point: PM3INT

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMINT

Cancelled by: N/A

Activates/calls: TGT50, PCHK, RNGSCHD

Cancels: None

Input arguments: None

Common variables: PV**

Files created/changed: None

Files referenced: None

PROCEDURE PM3INT

HANDOFF TO BE SCORED?

START TGT50

TURN TO FINAL TO BE SCORED?

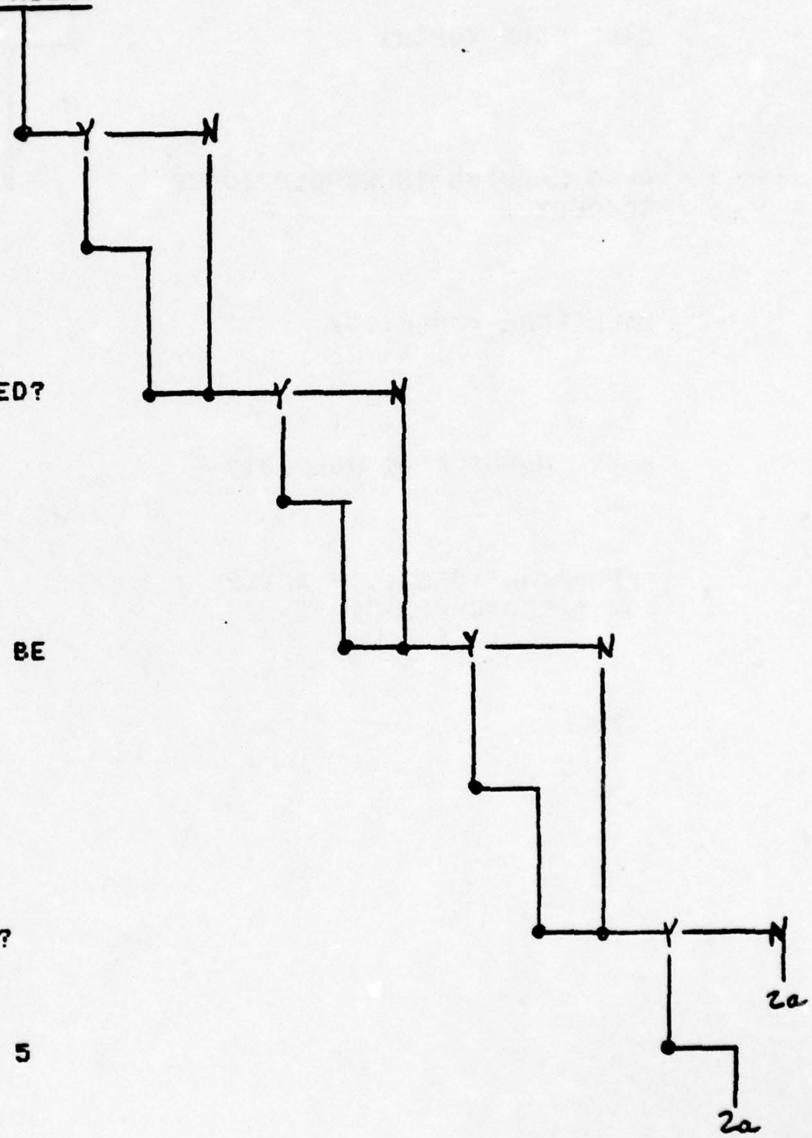
CALL PCHK (OMCK...)

APPROACHING GLIDEPATH TO BE SCORED?

CALL PCHK (OMCK,....)

RANGE CALLS TO BE SCORED?

CALL PCHK (OMCK,....) FOR 5 THROUGH 1 MILES



PM3INT FLOWCHART (SHEET 1 OF 2)

DECISION HEIGHT TO BE
SCORED?

CALL PCHK (OMCK)

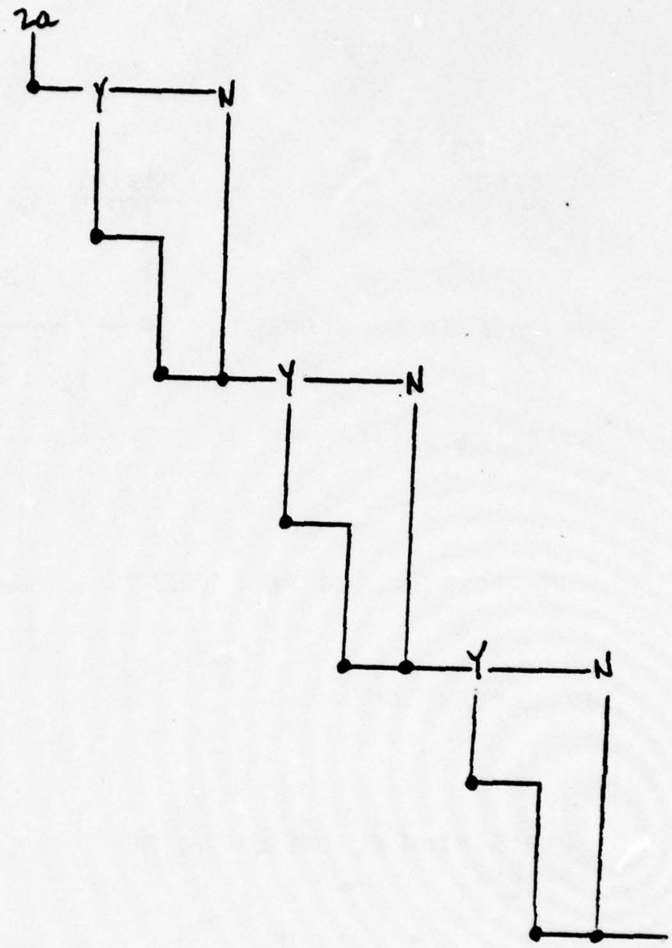
OVER LANDING THRESHOLD TO BE
SCORED?

CALL PCHK (OMCK...)

SCORE HANDOFF OR ROLLOUT?

CALL PCHK (OMCK...) AFTER
LOW APPROACH, ETC.

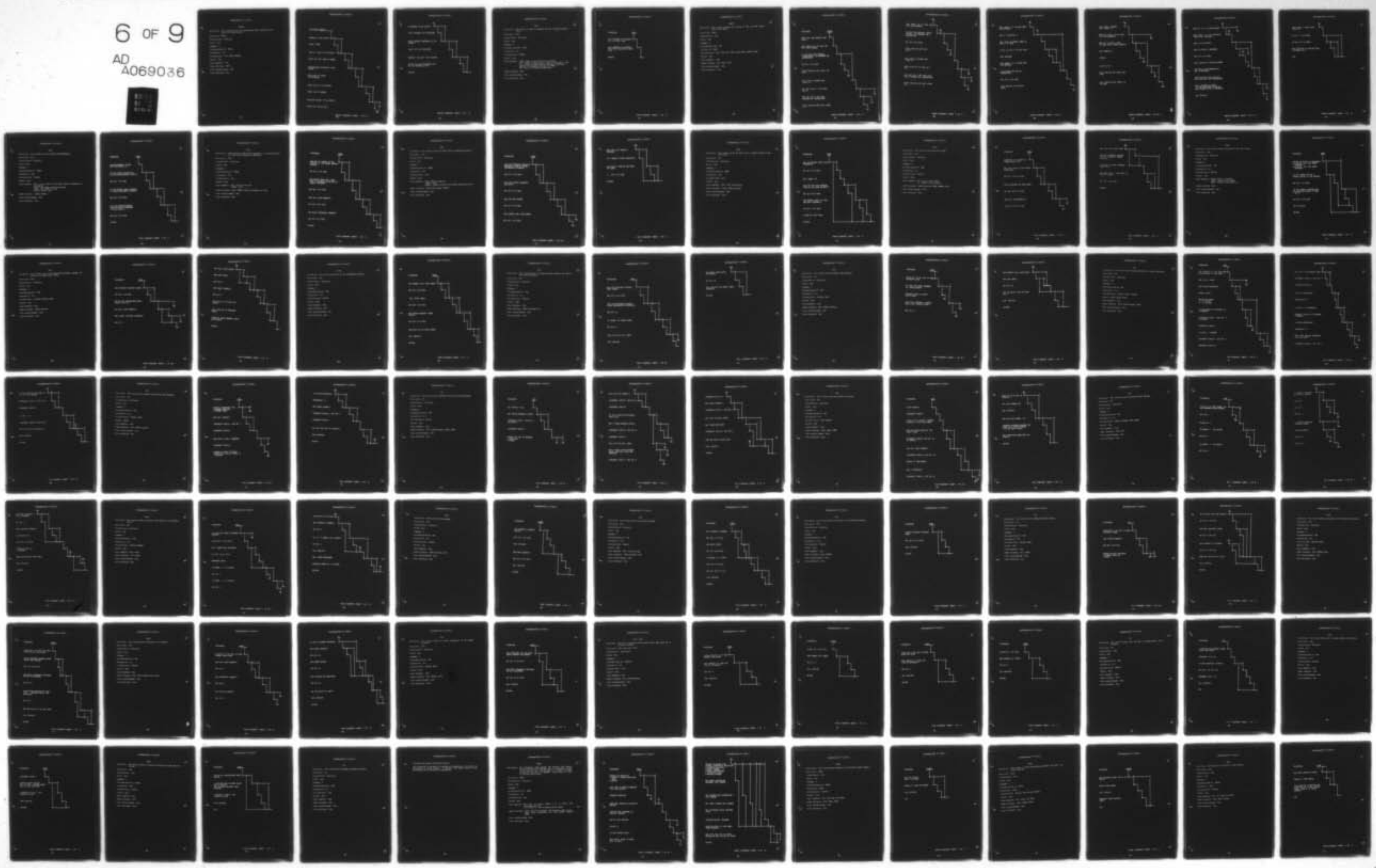
RETURN



PM3INT FLOWCHART (SHEET 2 OF 2)

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM. (U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

6 OF 9
AD
A069036



PMDINT

Description: This routine sets up the range-related model controller advisories for demonstrations.

Entry point: PMDINT

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMINT

Cancelled by: N/A

Activates/calls: PUT, TGT50, RNGSCHD

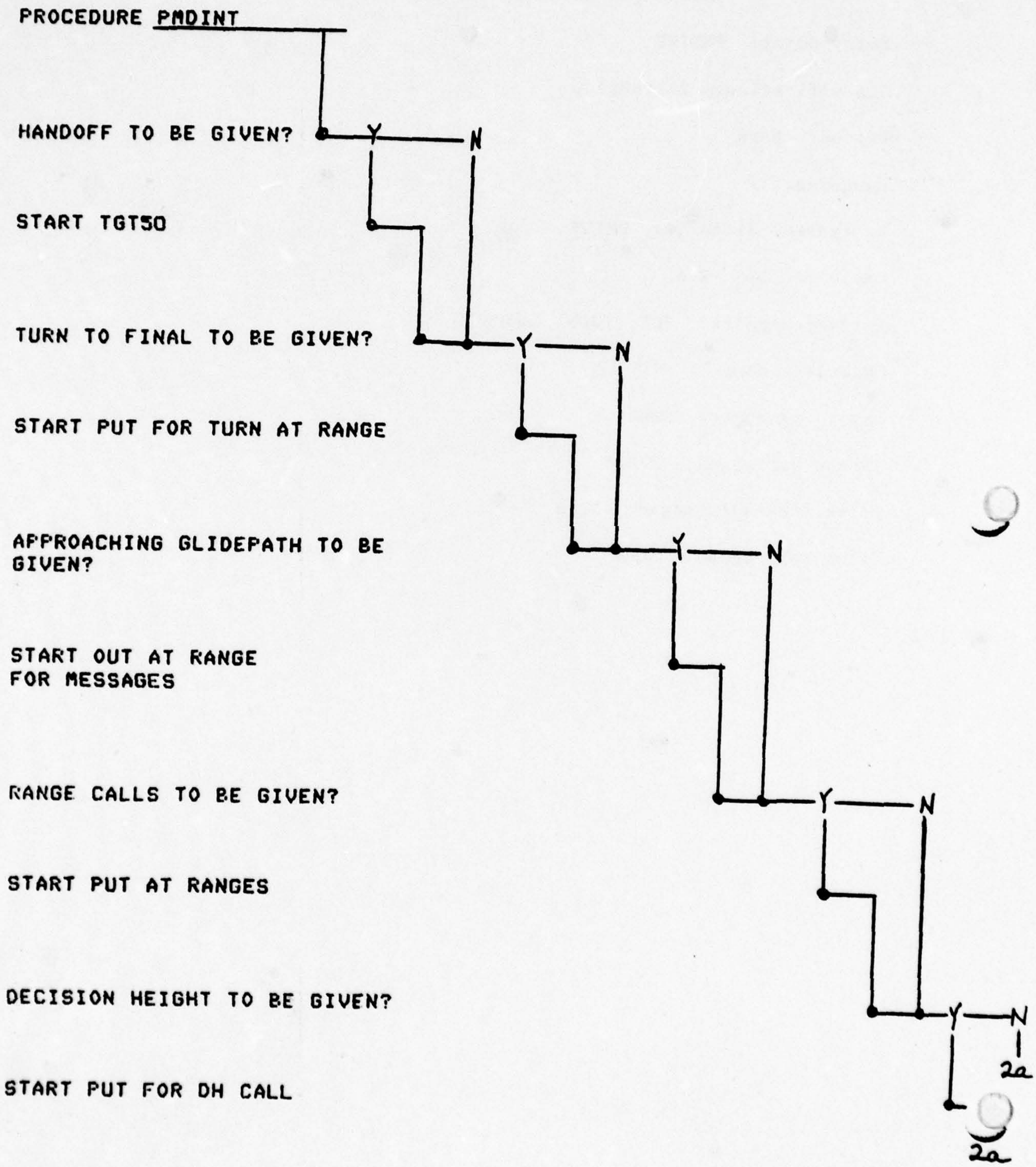
Cancels: None

Input arguments: None

Common variables: PFS**

Files created/changed: None

Files referenced: None



PMDINT FLOWCHART (SHEET 1 OF 2)

CLEARANCE TO BE GIVEN?

START RNGSCHD FOR PROCEDURE

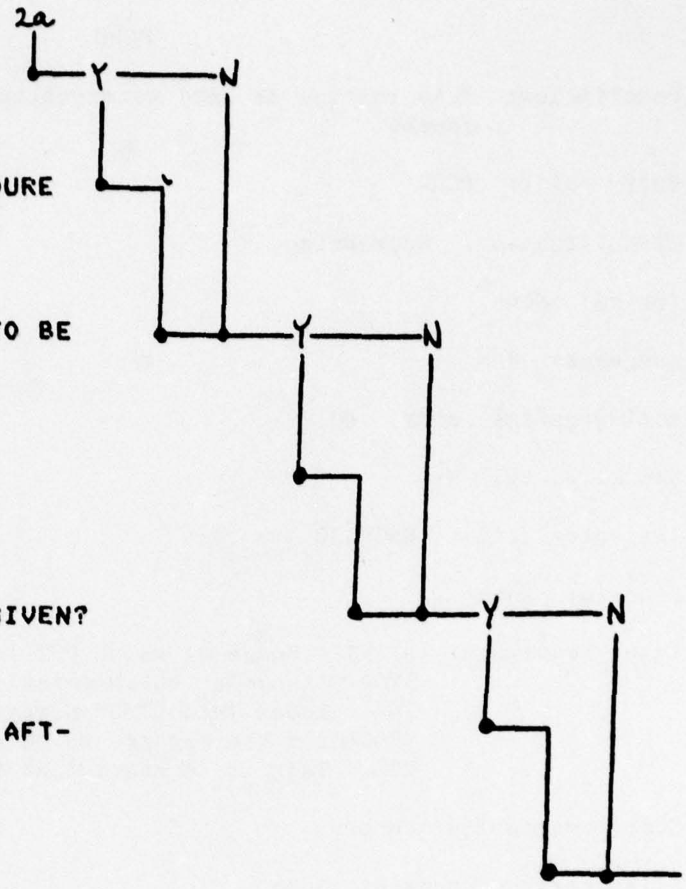
OVER LANDING THRESHOLD TO BE GIVEN?

START PUT FOR PROCEDURE

HANDOFF, ROLLOUT TO BE GIVEN?

START PUT FOR PROCEDURE AFTER LOW APPROACH, ETC.

RETURN



PMDINT FLOWCHART (SHEET 2 OF 2)

PCHK

Description: This routine is used to establish the set of omitted advisory checks.

Entry point: PCHK

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMINT

Cancelled by: N/A

Activates/calls: RNGSCHD

Cancels: None

Input arguments: START — Range at which PUT is to be started
STOP — Range at which omission check is to be made by OMCK
IDX — Index into CTMSG array of controller advisories
MESSAGE — The message to be set into CTMSG
TSK — Task to be started at STOP time

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE **PCHK**

CALL RNGSCHD TO SCHEDULE
PUT AT START TIME

CALL RNGSCHD TO SCHEDULE
SPECIFIED ROUTINE AT STOP
TIME

RETURN

PCHK FLOWCHART (SHEET 1 OF 1)

HRPMS

Description: This routine oversees the scoring of PV01 and PV02 during phase 2 and replay.

Entry point: HRPMS

Classification: Task

Period: None

Language: F

Activated/called by: PMS

Cancelled by: HRPMS

Activates/calls: P01A, P01B, P01C, P02A, P02B, RDACT, EXPLAIN, HRP1

Cancels: None

Input arguments: None

Common variables: PV01, PV02, KYIC4

Files created/changed: None

Files referenced: None

PROCEDURE

SAVE THE TIME HANDOFF WAS GIVEN

CALL RDACT (3) TO FIND THE NEXT SUS MESSAGE

IS THE NEXT SUS MESSAGE "ACKNOWLEDGE" OR MESSAGE NOT UNDERSTOOD?

SET BIT 3 OF PVO1

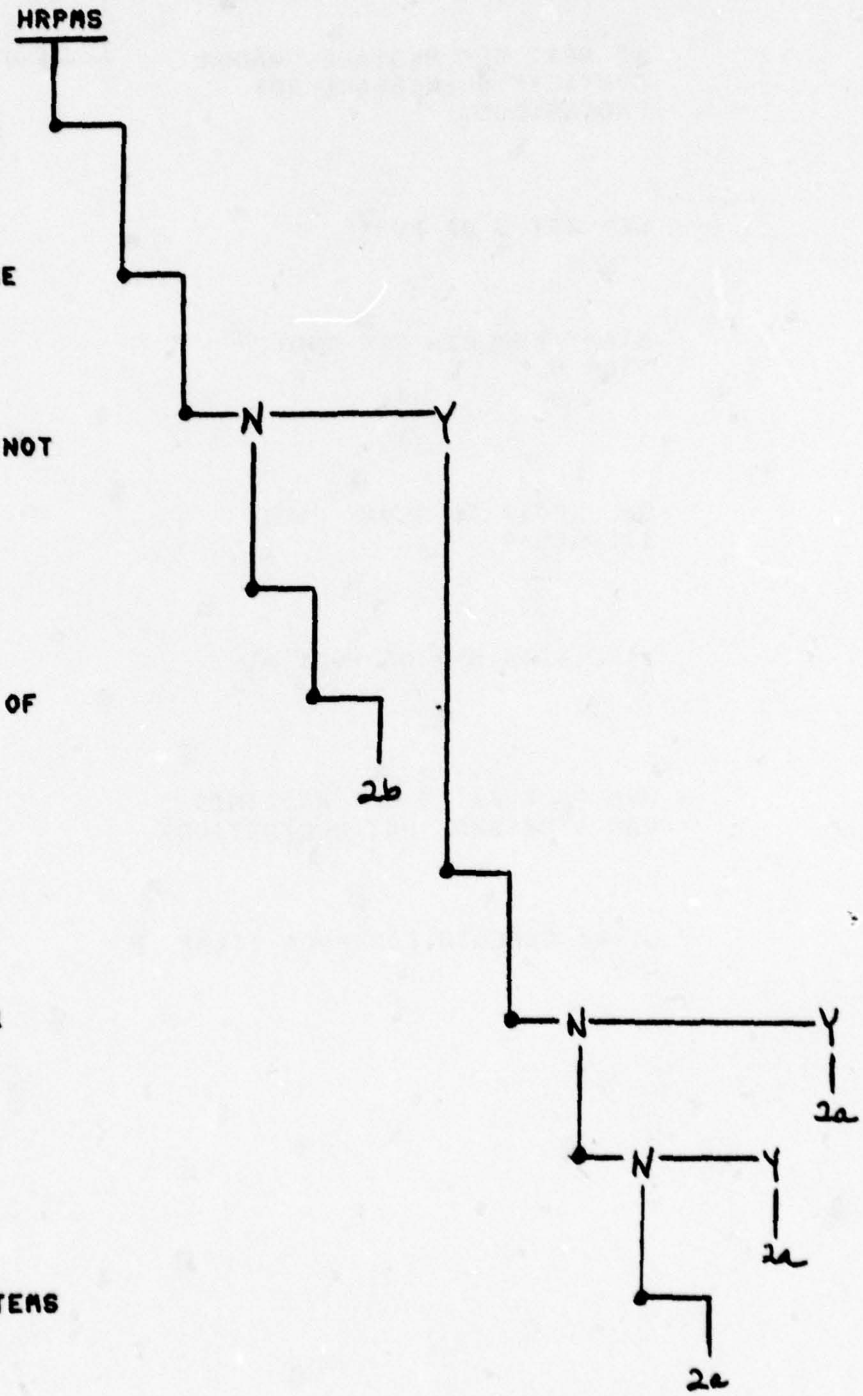
START EXPLAIN FOR ITEM 3 OF PVO1

CALL PO1A TO SCORE PVO1 ITEMS 1-4

WAS THE FIELD 1-4 OF PVO1 ALL 0?

WAS ONLY BIT 3 SET AND MESSAGE NOT UNDERSTOOD?

START EXPLAIN FOR PVO1 ITEMS 1-4



HRPMS FLOWCHART (SHEET 1 OF 6)

CALL RDACT (3) TO FIND THE NEXT SUS MESSAGE

IS NEXT SUS MESSAGE "RADAR CONTACT" OR MESSAGE NOT UNDERSTOOD?

SET BIT 5 OF PVO1

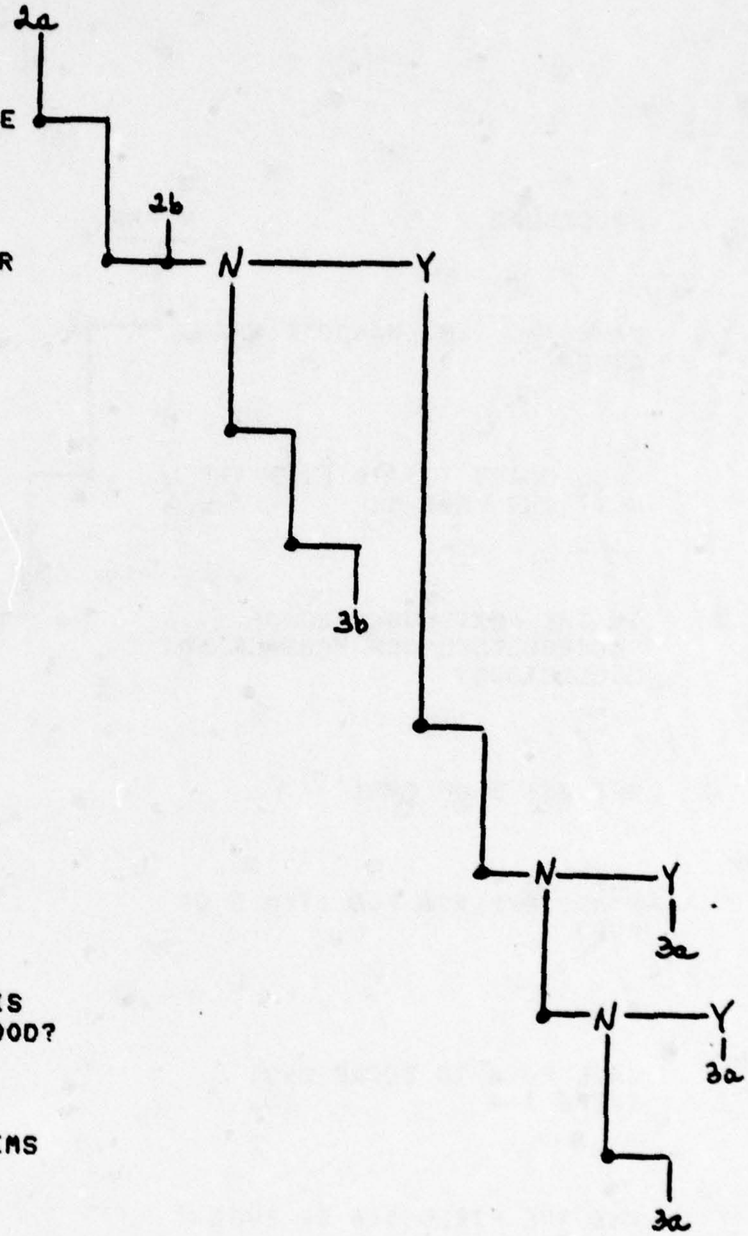
START EXPLAIN FOR PVO1 ITEM 5

CALL PO1B TO SCORE PVO1 ITEMS 5-9

WERE BITS 5-9 OF PVO1 ALL 0?

WAS ONLY BIT 5 SET AND THIS WAS A MESSAGE NOT UNDERSTOOD?

START EXPLAIN FOR PVO1 ITEMS 5-9



HRPMS FLOWCHART (SHEET 2 OF 6)

CALL RDACT (3) TO READ NEXT
SUS MESSAGE

WAS IT "GIVE ME..."

CALL P01C TO SCORE ITEMS 10
AND 11 OF PV01

IS BIT 10 OR 11 OF PV01 SET?

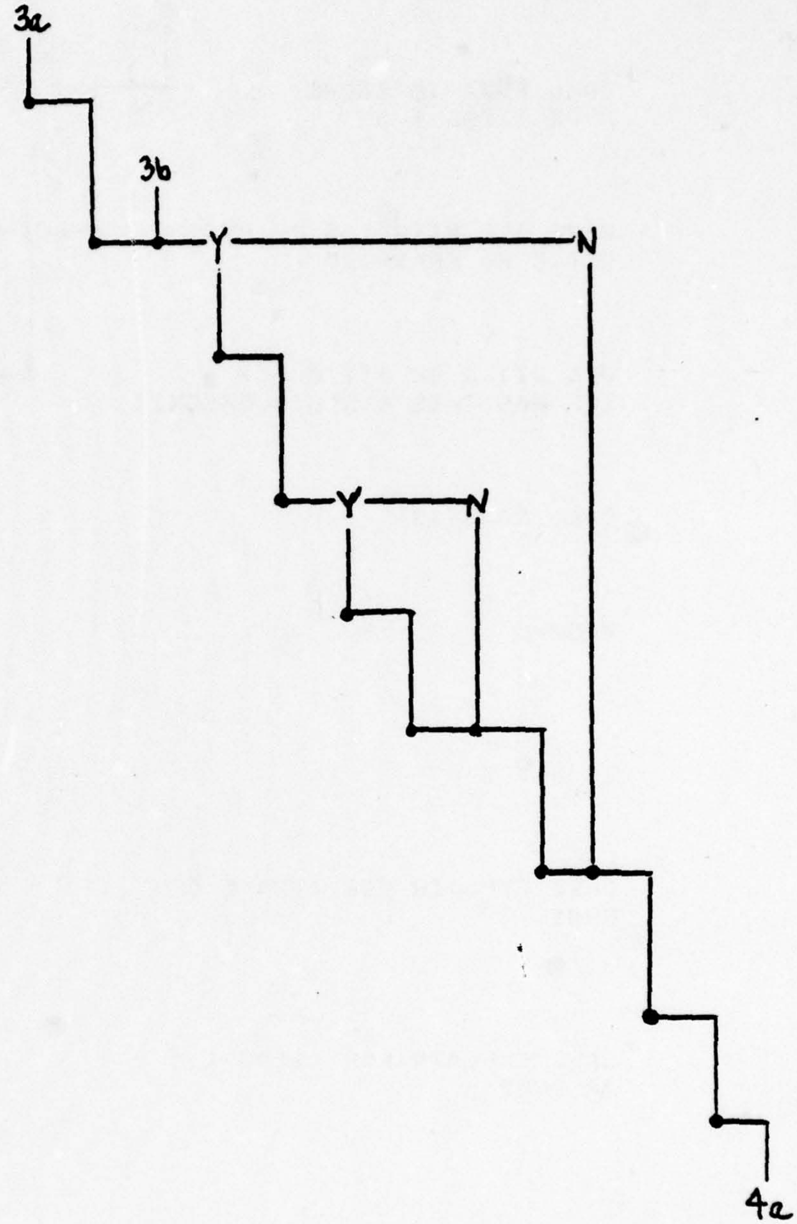
CALL EXPLAIN

CALL RDACT (3) TO READ NEXT
SUS MESSAGE

IS PATTERN ICS BUTTON
DESELECTED?

SET BIT 12 OF PV01

CALL EXPLAIN TO EXPLAIN
ERROR



HRPMS FLOWCHART (SHEET 3 OF 6)

CALL PO2A TO SCORE
PVO2 ITEMS 1-5

WERE ALL BITS 1-5 OF PVO2
O (IE NO ERRORS)?

WAS BIT 3 OR BIT 5 SET
(IE WAS THIS A RADIO CHECK)?

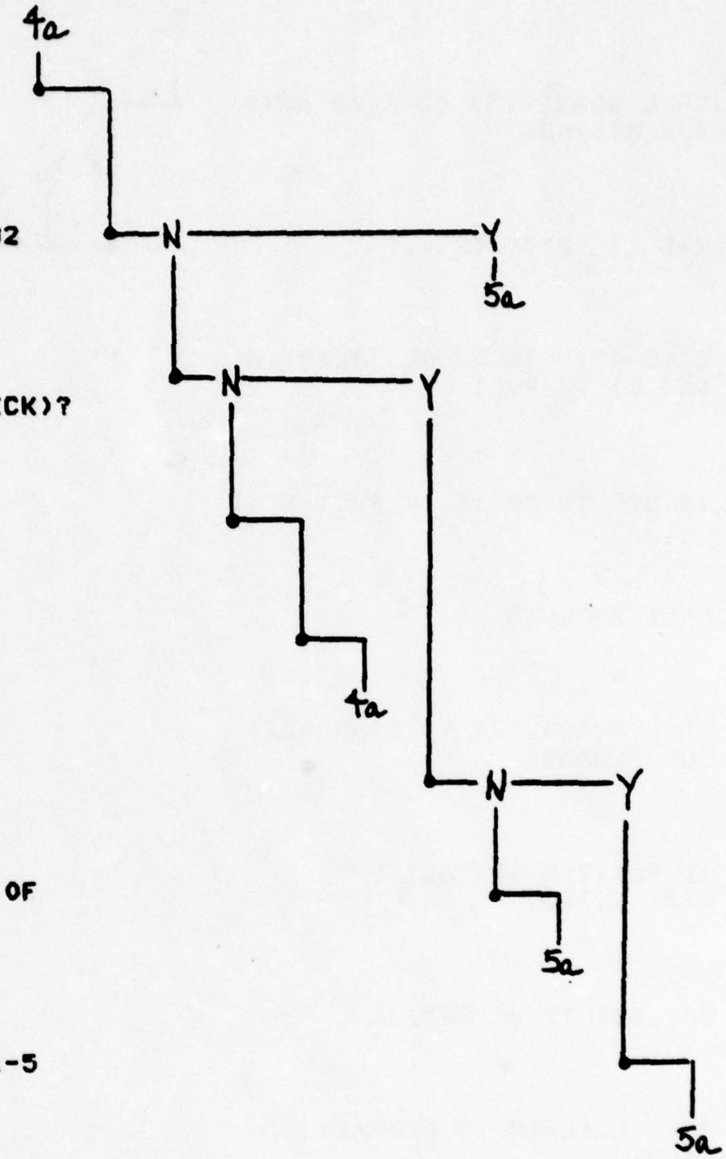
CALL RDACT(3)

PVO2=0

IS BIT 5 0?

CALL EXPLAIN FOR ITEM 5 OF
PVO2

CALL EXPLAIN FOR ITEMS 1-5
OF PVO2



HRPMS FLOWCHART (SHEET 4 OF 6)

WAS BIT 3 0, IE MIKE KEYED?

CALL RDACT (-1) TO FIND NEXT
THING THAT HAPPENED

WAS IT MIKE UNKEY?

WAS IT WITHIN 3 SECONDS?

SET BIT 6 OF PV02

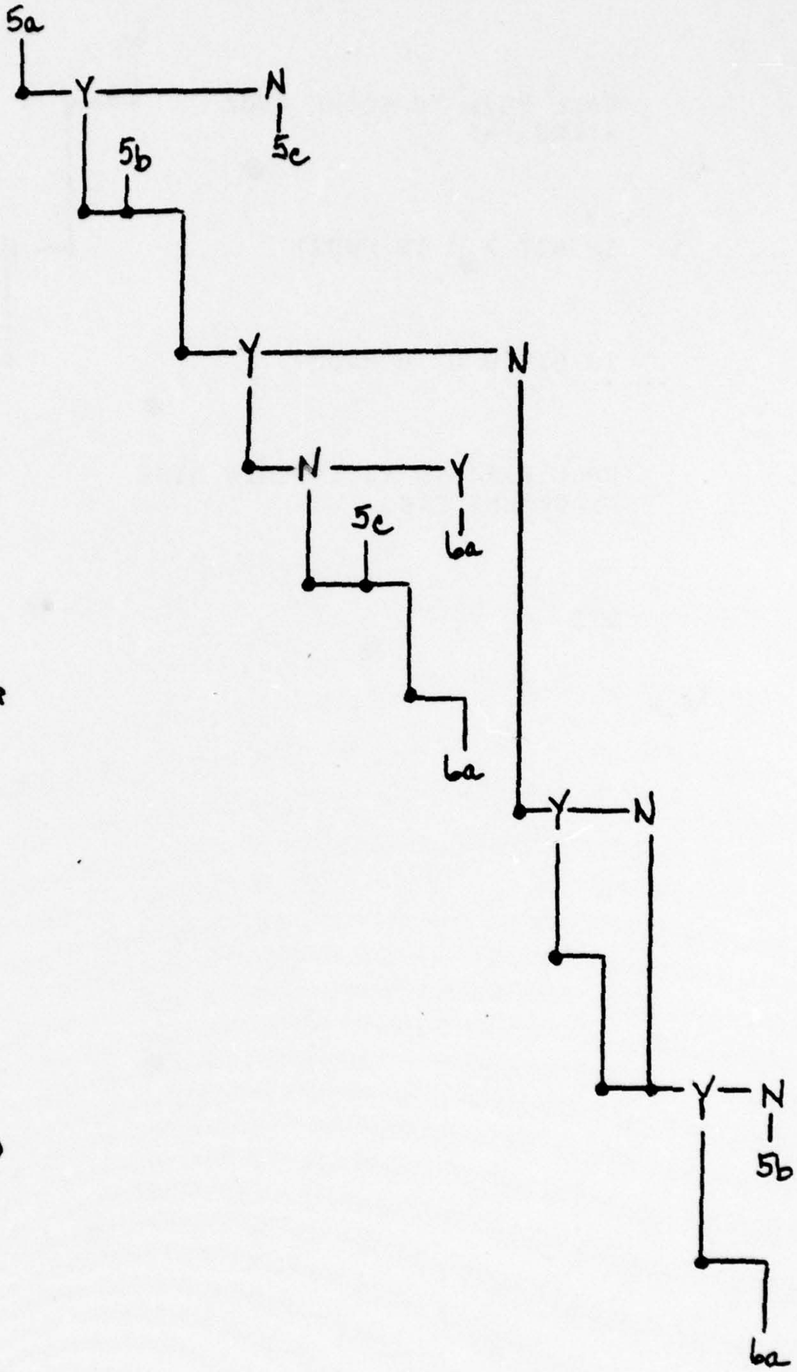
CALL EXPLAIN TO EXPLAIN ERROR

WAS THIS A SUS MESSAGE WE
HAD TO PROCESS?

SAVE POINTER INTO ACTIVITY
FILE FOR NEXT SUS PROCESSING

HAS 3 SECONDS ELAPSED?
(IE WAS WHATEVER ACTION WE
CAME ACROSS AFTER 3 SECONDS?)

CALL EXPLAIN



HRPMS FLOWCHART (SHEET 5 OF 6)

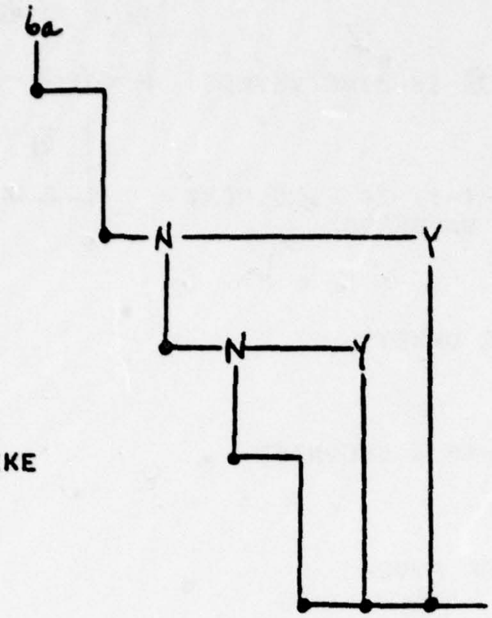
CALL P02B TO SCORE PV02
ITEMS 7-9

IS BIT 7 0 IN PV02?

IS BIT 8 0 IN PV02?

CALL EXPLAIN TO EXPLAIN MIKE
PLACEMENT ETC.

DIE



HRPMS FLOWCHART (SHEET 6 OF 6)

P01A

Description: This routine scores the handoff acknowledgement.

Entry point: P01A

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRPMS

Cancelled by: P01A

Activates/calls: EXPLAIN

Cancels: None

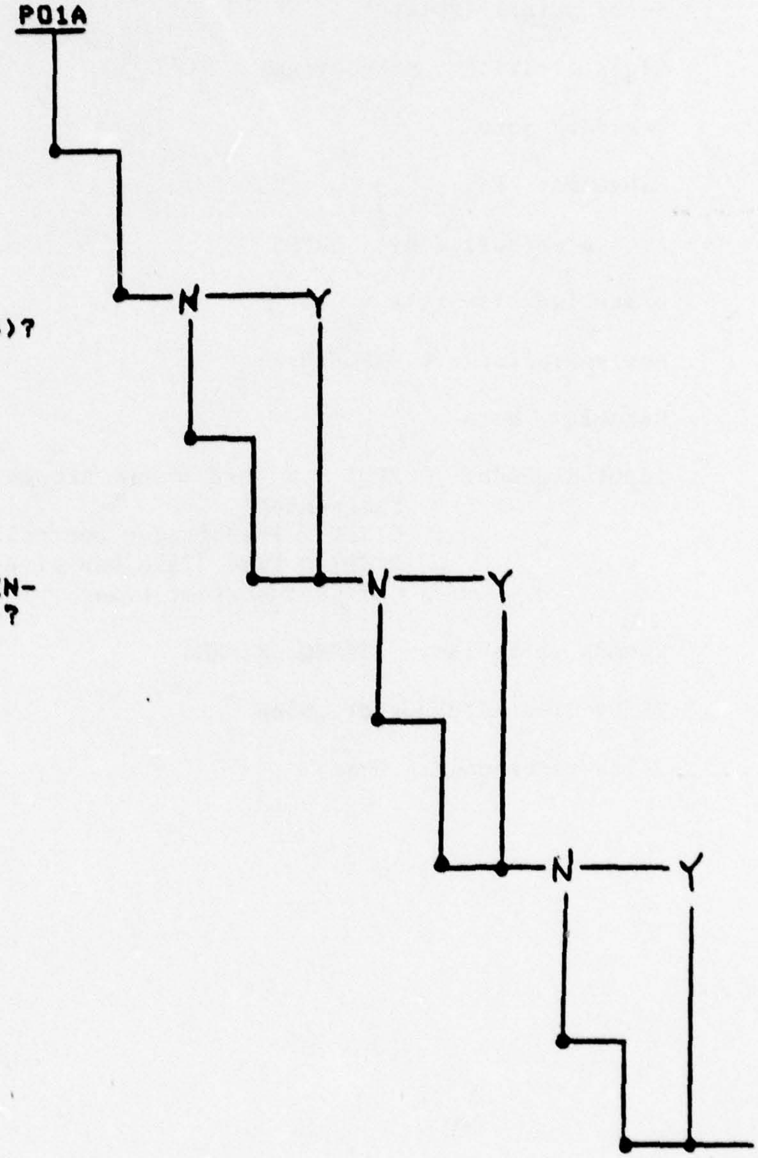
Input arguments: PV01 — a word whose bit settings indicate performance on the handoff
CTSPH — last feeder controller message
CTSPT — time CTSPH was given
CLTICK — current time

Common variables: CTFREQ, KYFREQ

Files created/changed: None

Files referenced: None

PROCEDURE PO1A



ACKNOWLEDGEMENT GIVEN,
BIT 3 OF PVO1 <- 0

IS THE FEEDER CONTROLLER
ICS BUTTON SELECTED (KYIC4)?

SET BIT 1 OF PVO1

IS THE PROPER RADIO FREQUEN-
CY BEING MONITORED (KYFRQ)?

SET BIT 2 OF PVO1

WAS THE ACKNOWLEDGEMENT
GIVEN WITHIN 10 SECONDS
(CLTICK,CTSPT)?

SET BIT 4 OF PVO1

RETURN

PO1A FLOWCHART (SHEET 1 OF 1)

P01B

Description: P01B scores the radar contact procedure. It is activated when the radar contact message is understood.

Entry point: P01B

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRPMS

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: PV01 — PV01 scoring word
CLTIME — time

Common variables: ACCS, CTFREQ, KYFRQ, SUS message call sign

Files created/changed: None

Files referenced: None

PROCEDURE

WAS 50% OF TARGET ON THE DISPLAY AT THE RADAR CONTACT REPORT?

SET BIT 6 OF PV01

WAS REPORT MADE NOT LATER THAN 10 SECONDS AFTER 50% OF TARGET APPEARED?

SET BIT 7 OF PV01

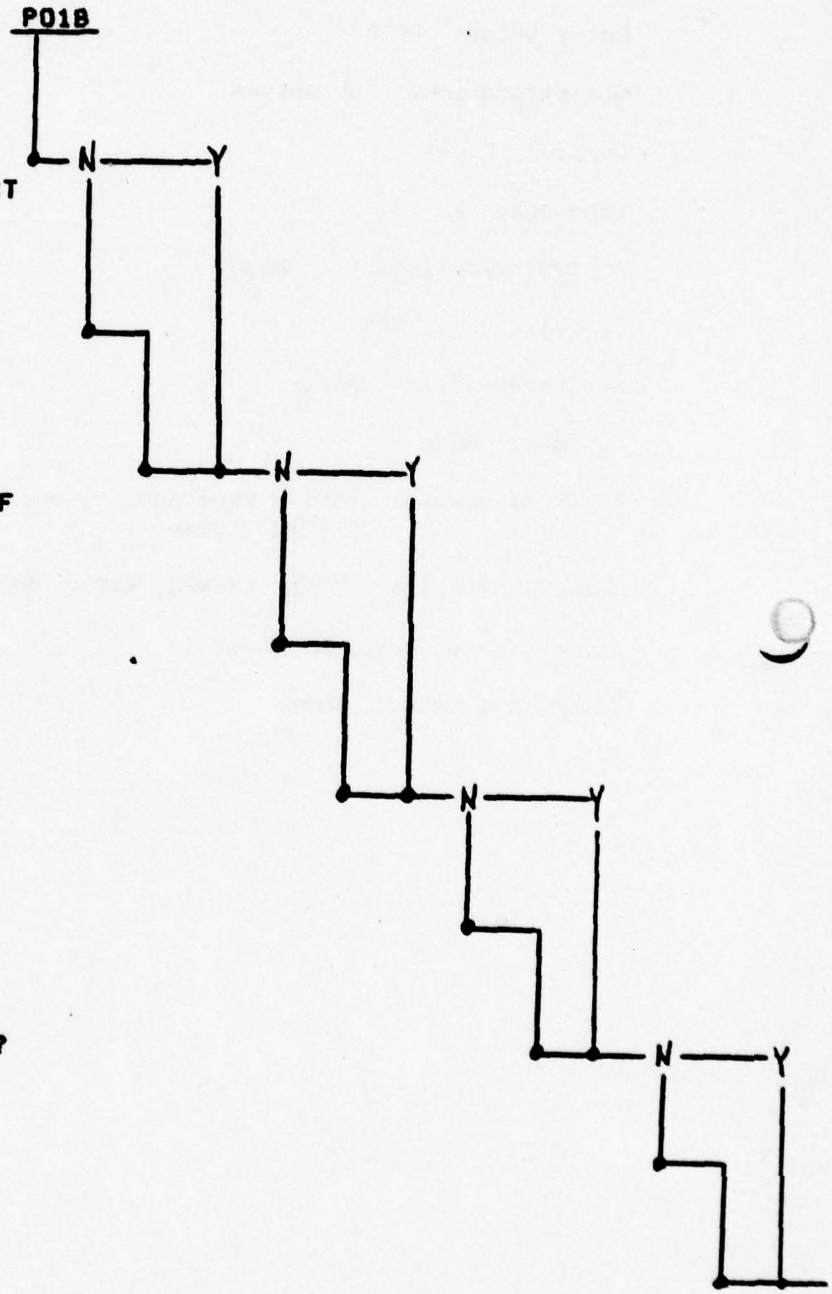
WAS CALL SIGN CORRECT?

SET BIT 8 OF PV01

WAS RADIO FREQUENCY CORRECT?

SET BIT 9 OF PV01

RETURN



P01B FLOWCHART (SHEET 1 OF 1)

P02A

Description: This routine scores the radio check for timing and accuracy.

Entry point: P02A

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRPMS

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: PV02 — PV02 scoring word
CLTIME — time
LWHEEL — Logical indicator that wheels message was given

Common variables: KYFRQ, SUS message, CLTGT50

Files created/changed: None

Files referenced: None

PROCEDURE

WAS THIS MESSAGE WITHIN 30
SECONDS OF 50% OF TARGET
APPEARING ON SCREEN?

SET BIT 1 OF PV02

WAS THE PROPER FREQUENCY
SELECTED?

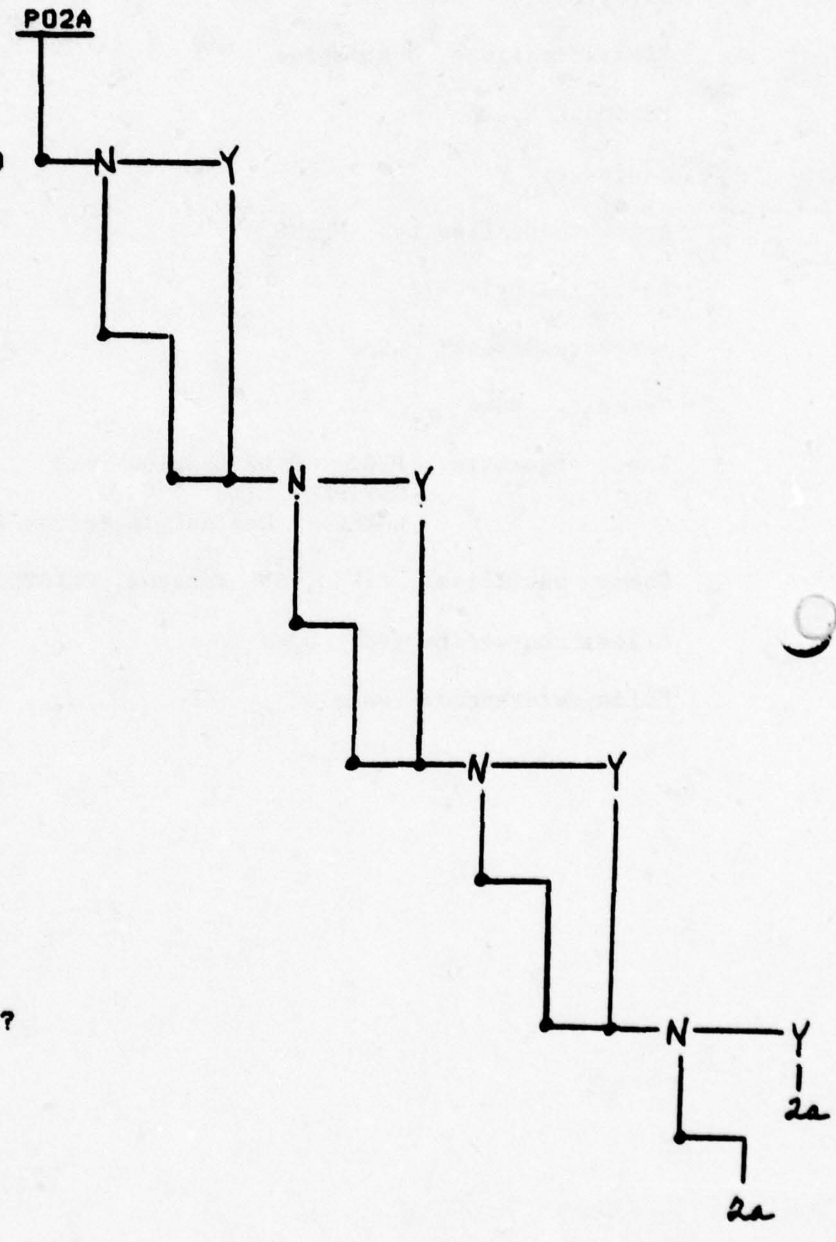
SET BIT 2 OF PV02

WAS THE MIKE KEYED?

SET BIT 3 OF PV02

WAS CORRECT CALL SIGN USED?

SET BIT 4 OF PV02



P02A FLOWCHART (SHEET 1 OF 2)

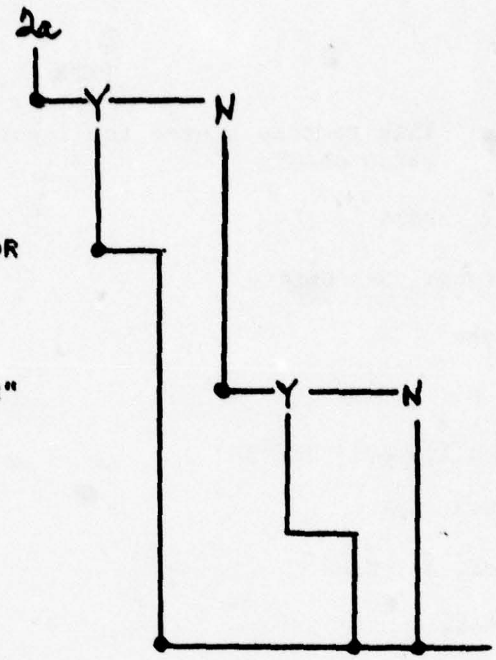
WAS THIS THE "WHEELS" MESSAGE?

SET "WHEELS" GIVEN INDICATOR

WAS THIS A "HOW DO YOU HEAR" OR TURN?

0 BIT 5 OF PV02

RETURN



PO2A FLOWCHART (SHEET 2 OF 2)

P02B

Description: This routine scores the input level or speech quality of the radio check.

Entry point: P02B

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRPMS

Cancelled by: N/A

Activates/calls: None

Cancels: None

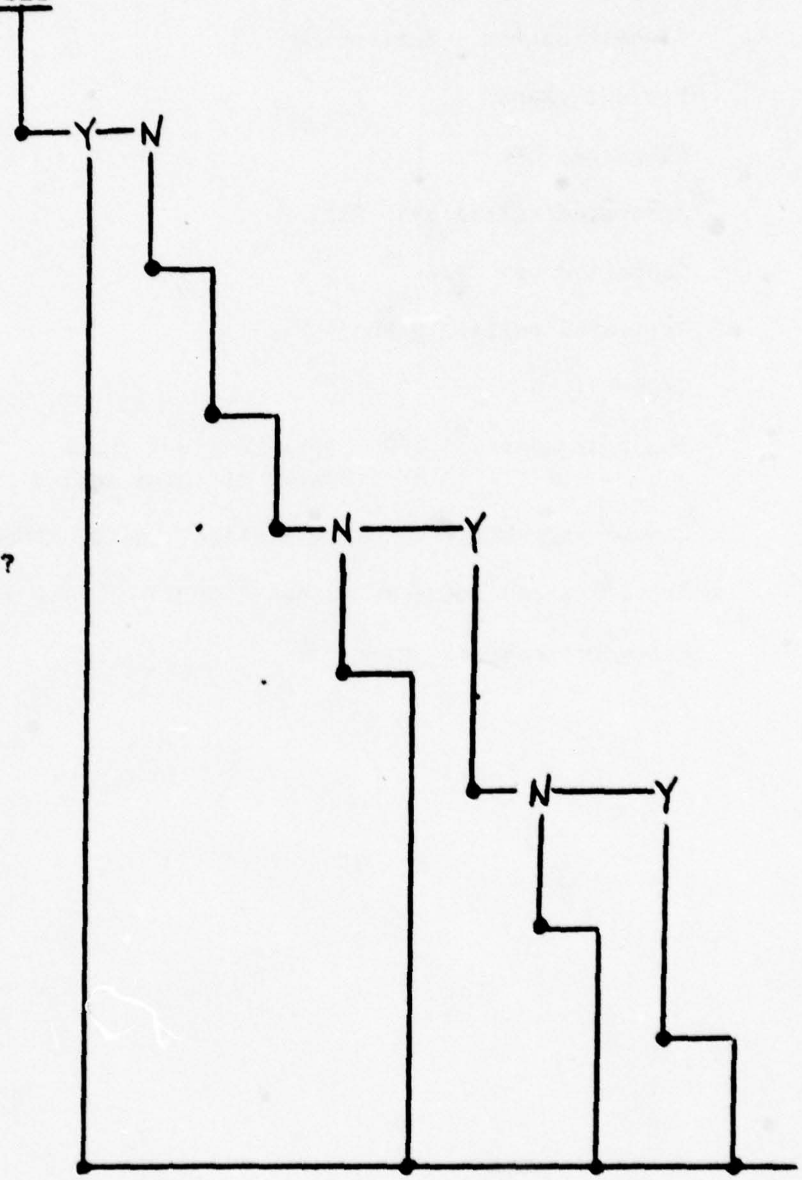
Input arguments: PV02 — PV02 scoring word

Common variables: KYLVL, SHUSH variables

Files created/changed: None

Files referenced: None

PROCEDURE P02B



WAS THE SPEECH LEVEL OVER THRESHOLD?

SET BIT 7 OF PVO2

CALL RDACT (3)

WAS THE NEXT SUS MESSAGE "HOW DO YOU HEAR ME NOW?"?

SET BIT 8 OF PVO2

WAS SPEECH LEVEL OF THIS ADVISORY ADEQUATE?

SET BIT 7 OF PVO2

CLEAR BIT 8 OF PVO2

RETURN

P02B FLOWCHART (SHEET 1 OF 1)

P03A

Description: This routine scores the turn to final.

Entry point: P03A

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRP1

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: IPH — Turn to final phase
NT — Number of turns scored

Common variables: SHUSH variables, ENWSP, ENWHDG, PV03

Files created/changed: None

Files referenced: None

PROCEDURE P03A

INCREMENT THE NUMBER OF
TURNS GIVEN (=NT)

WAS THIS TURN IN THE PROPER
DIRECTION?

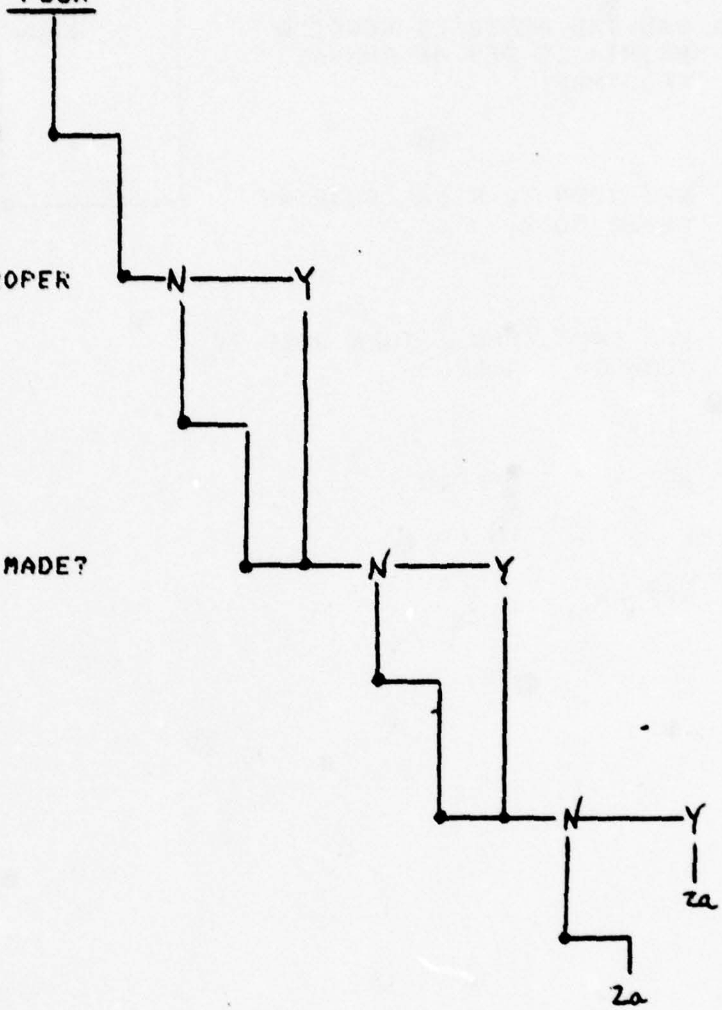
SET BIT 7+NT OF PV03

WAS ALLOWANCE FOR WIND MADE?

SET BIT 8+NT OF PV03

WAS CALL SIGN CORRECT?

SET BIT 9+NT OF PV03



P03A FLOWCHART (SHEET 1 OF 2)

WAS THIS THE THIRD TURN?

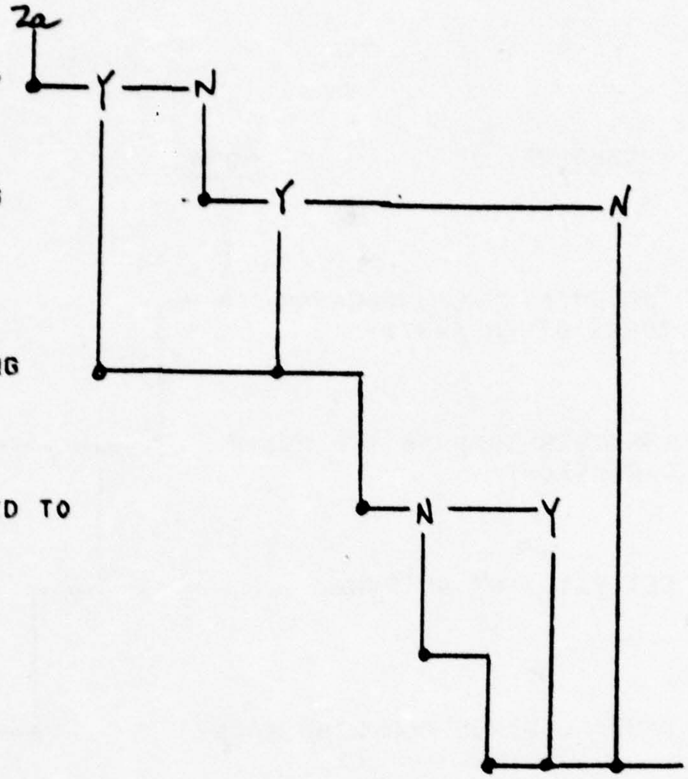
WAS THE ASSIGNED HEADING WITHIN 10 DEG OF RUNWAY HEADING?

SET TURN TO FINAL SCORING PHASE TO 2

WAS MORE THAN 1 TURN USED TO TURN TO FINAL?

SET BIT 3 OF PV03

RETURN



P03A FLOWCHART (SHEET 2 OF 2)

P03B

Description: This routine scores the quality of the turn to final.

Entry point: P03B

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: MILES — Miles to touchdown
AFOOT — distance from azimuth
EFOOT — distance from elevation

Common variables: None

Files created/changed: None

Files referenced: None

PROCEDURE

BRANCH ON MILES TO TOUCHDOWN
1-6 MILES, OR 3 FOR SHORT
APPROACH
2-5 MILES, OR 2 FOR SHORT
APPROACH

IS THE TARGET WITHIN 2
TARGET WIDTHS OF THE CURSOR?

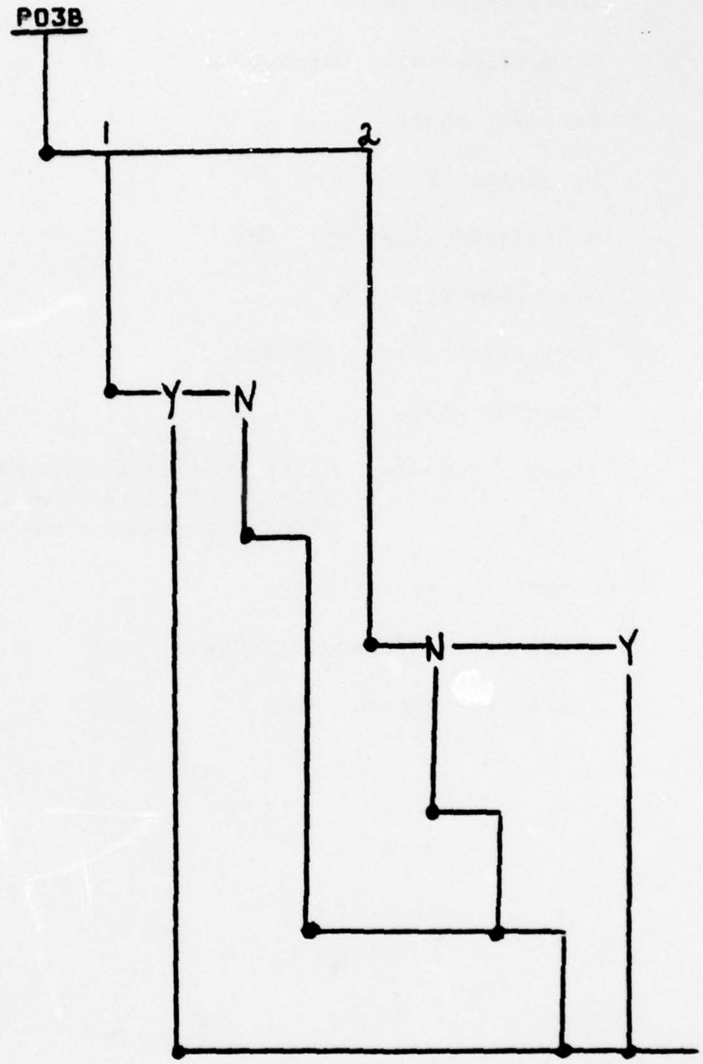
SET BIT 1 OF PV03

IS THE TARGET INTERCEPTING
THE AZIMUTH CURSOR IN ZONE
1 OR 2?

SET BIT 2 OF PV03

CALL EXPLAIN

RETURN



P03B FLOWCHART (SHEET 1 OF 1)

P04A

Description: This routine scores the approaching glidepath message and schedules the begin descent check.

Entry point: P04A

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: TIMSCHD, EXPLAIN, RDACT

Cancels: None

Input arguments: None

Common variables: SHUSH variables

Files created/changed: None

Files referenced: None

PROCEDURE

WAS ADVISORY ALREADY GIVEN?

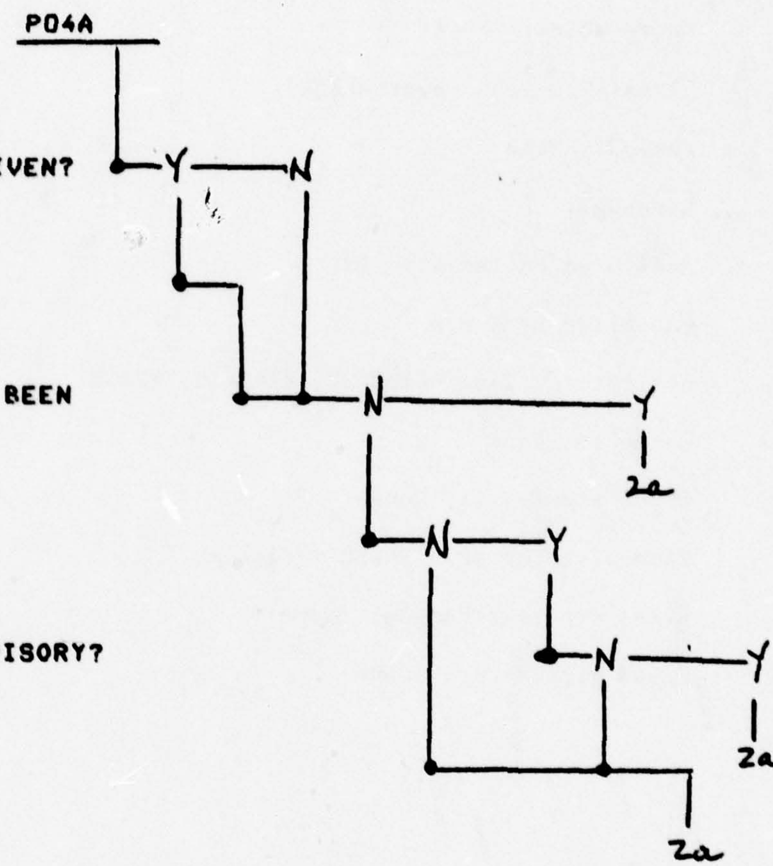
SET BIT 4 OF PV04

HAS DO NOT ACKNOWLEDGE BEEN GIVEN? (BIT 5)

WAS CALL SIGN CORRECT?

WAS "OVER" THE NEXT ADVISORY?

SET BIT 1



P04A FLOWCHART (SHEET 1 OF 2)

WAS CALL SIGN GIVEN?

WAS OVER USED?

SET BIT 2

WAS RANGE CORRECT?

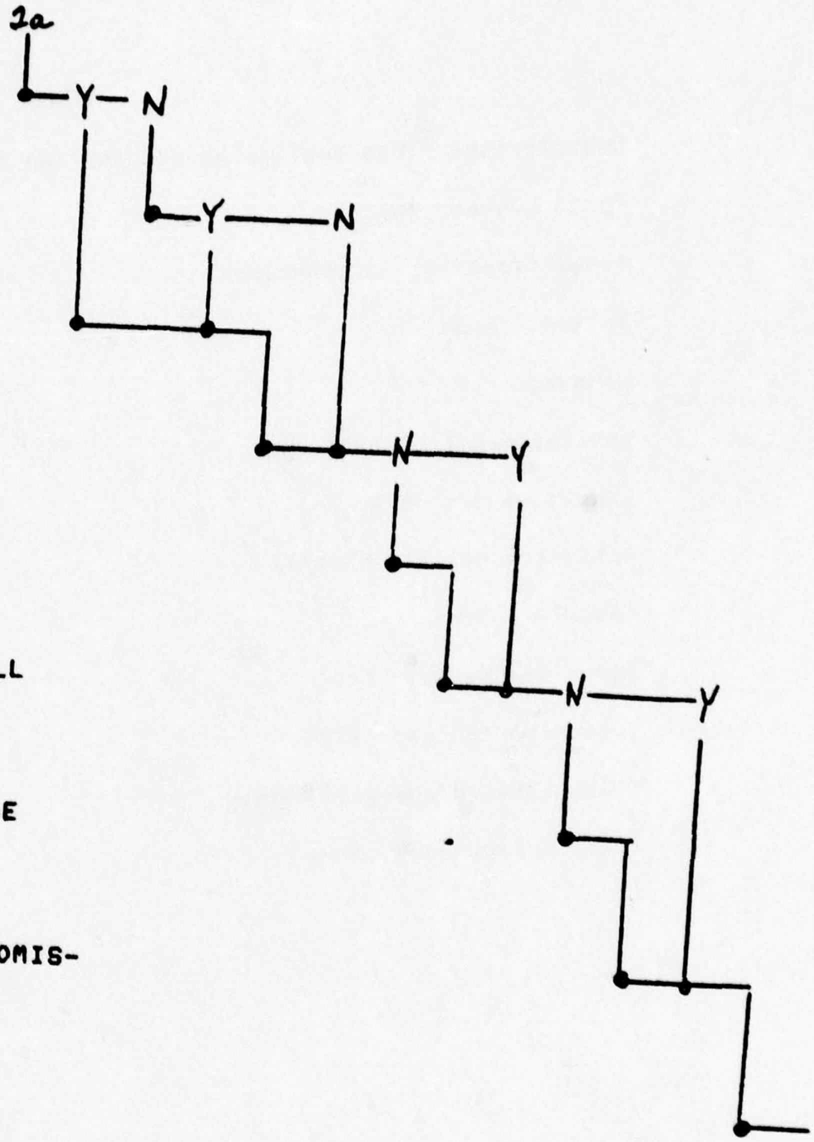
SET BIT 3

ARE BITS 0-4 OF PV04 ALL
CLEAR?

CALL EXPLAIN TO DESCRIBE
ERROR

SCHEDULE BEGIN DESCENT OMIS-
SION CHECK

RETURN



P04A FLOWCHART (SHEET 2 OF 2)

P04B

Description: This routine scores the "do not acknowledge" advisory.

Entry point: P04B

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

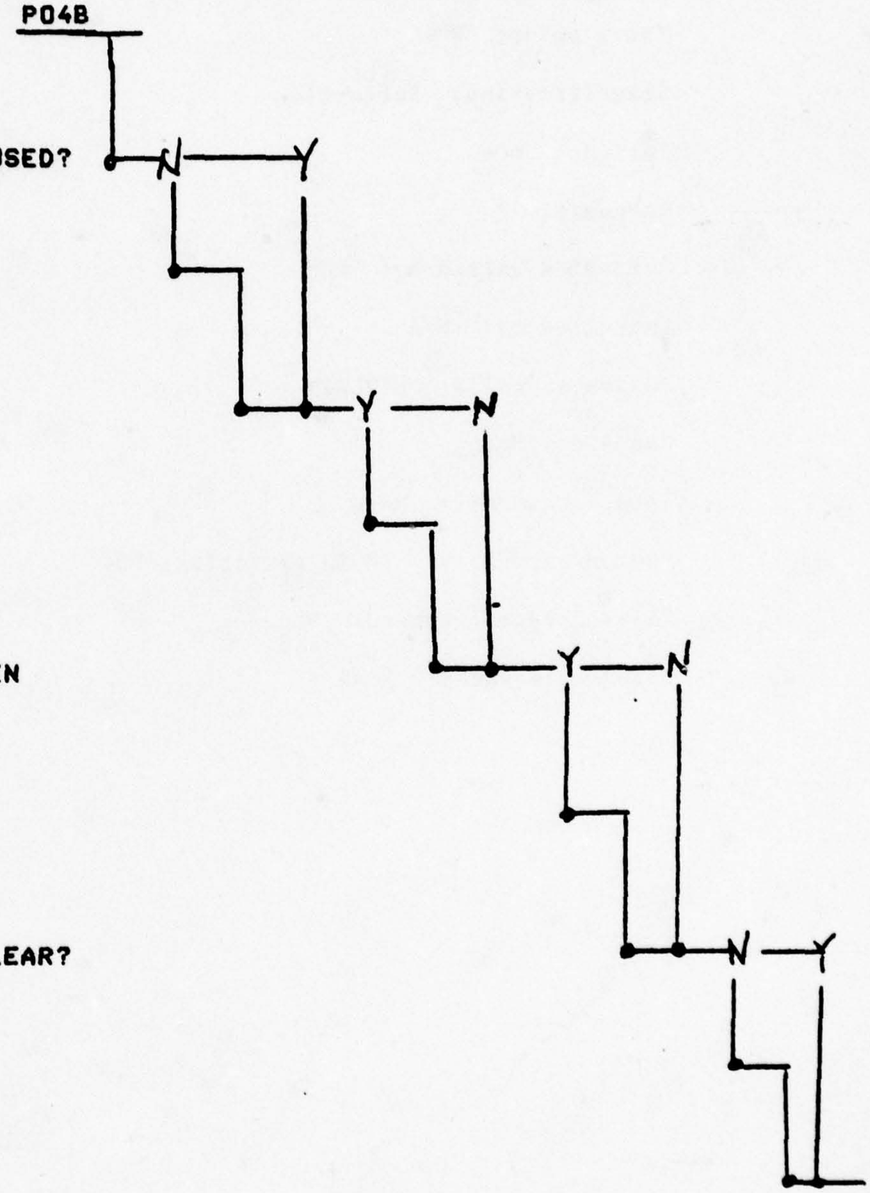
Input arguments: None

Common variables: PV04

Files created/changed: None

Files referenced: None

PROCEDURE P04B



WAS CORRECT CALL SIGN USED?

SET BIT 6 OF PV04

WAS "OVER" USED?

SET BIT 7 OF PV04

HAS "BEGIN DESCENT" BEEN GIVEN?

SET BIT 8 OF PV04

ARE BITS 6-8 OF PV04 CLEAR?

CALL EXPLAIN

RETURN

P04B FLOWCHART (SHEET 1 OF 1)

P04C

Description: This routine scores the begin descent advisory and detects wheel check omissions.

Entry point: P04C

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

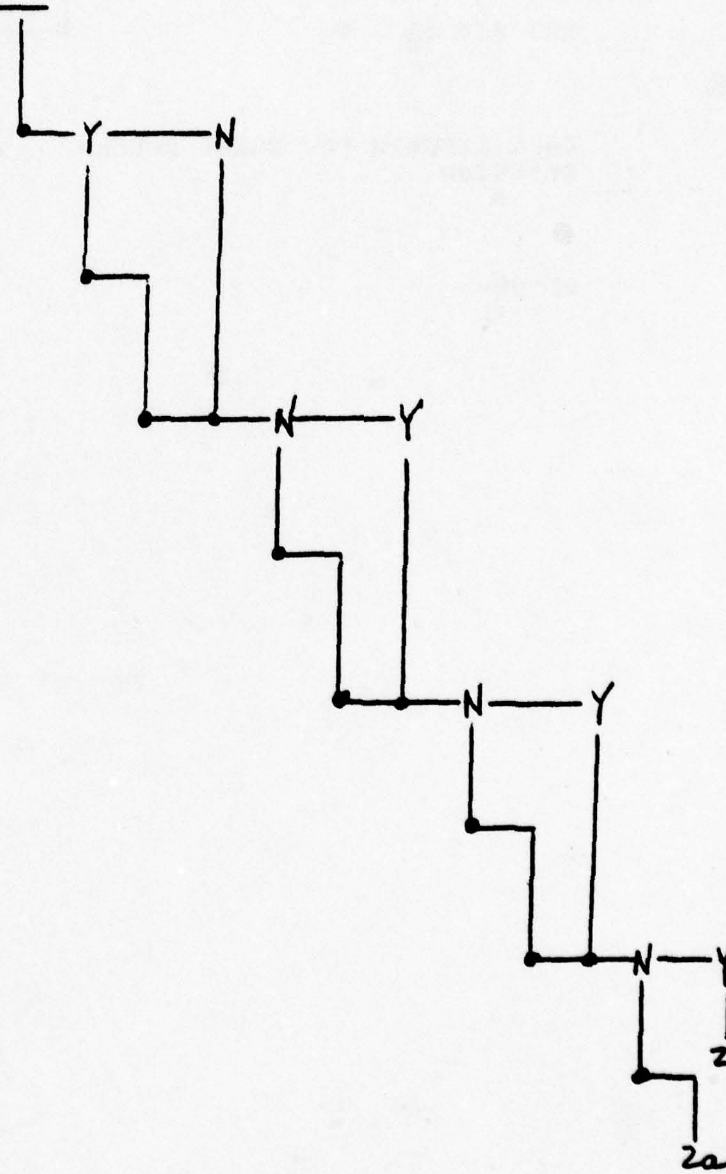
Input arguments: None

Common variables: SHUSH variables, P04

Files created/changed: None

Files referenced: None

PROCEDURE P04C



HAS THIS ADVISORY ALREADY
BEEN GIVEN?

SET BIT 12 OF PV04

HAS 10-30 SECONDS ELAPSED
SINCE APPROACHING GLIDEPATH?

SET BIT 10

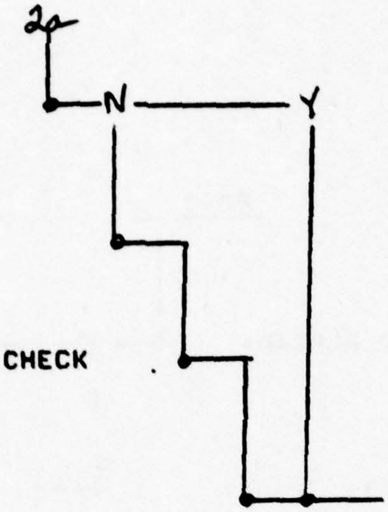
IS TARGET IN PROPER ZONE?

SET BIT 11

ARE BITS 10-12 ALL ZERO?

CALL EXPLAIN

P04C FLOWCHART (SHEET 1 OF 2)



HAS WHEEL CHECK BEEN SATISFIED?

SET BIT 13

CALL EXPLAIN FOR WHEEL CHECK OMISSION

RETURN

P04C FLOWCHART (SHEET 2 OF 2)

P04D

Description: This routine scores the wheels check advisory.

Entry point: P04

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, P02A

Cancels: None

Input arguments: None

Common variables: PV04, SHUSH variables

Files created/changed: None

Files referenced: None

PROCEDURE

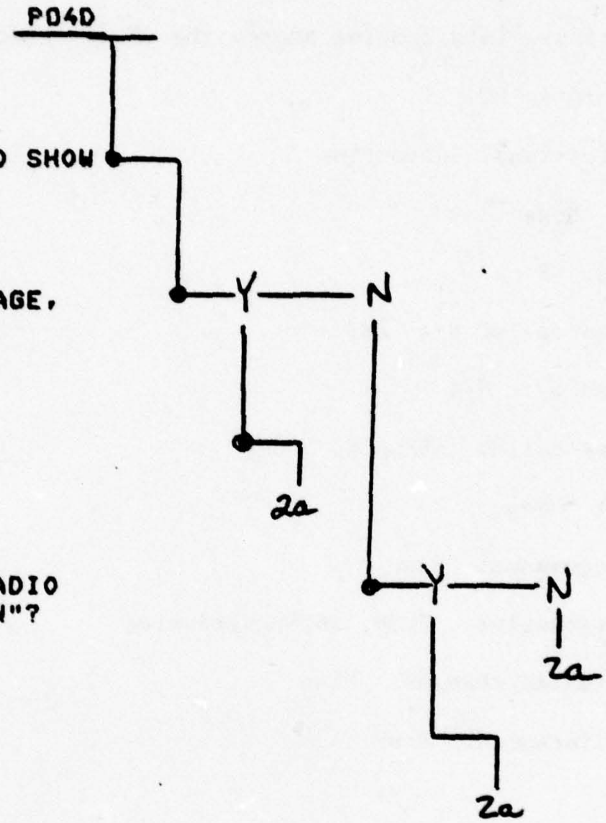
CLEAR BIT 13 OF PV04 TO SHOW
ADVISORY GIVEN

IS THIS THE FIRST MESSAGE,
IE. RADIO CHECK?

SCHEDULE P02A TO SCORE
RADIO CHECK

DID PILOT RESPOND TO RADIO
CHECK WITH "WHEELS DOWN"?

SET BIT 14



P04D FLOWCHART (SHEET 1 OF 2)

WAS CORRECT CALL SIGN USED?

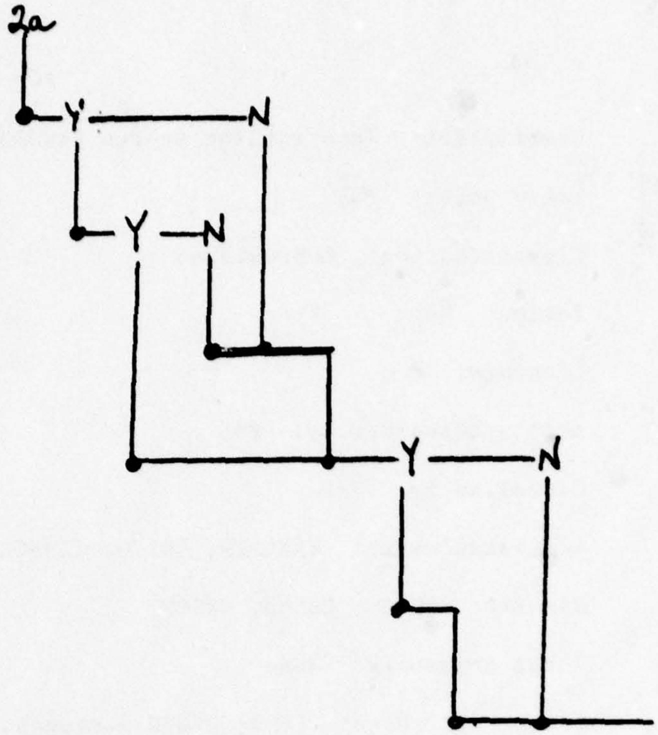
WAS OVER USED?

SET BIT 15

ARE ANY BITS 13-15 OF PV04
SET?

CALL EXPLAIN

RETURN



PO4D FLOWCHART (SHEET 2 OF 2)

P05

Description: This routine scores heading advisories on normal approaches.

Entry point: P05

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, CK120, TIMSCHD

Cancels: CK120, CKZN3, CKCOR

Input arguments: None

Common variables: PV05, SHUSH messages, PFHDG

Files created/changed: None

Files referenced: None

PROCEDURE P05

SET PVO5(0) 0. BIT SETTINGS
BELOW REFER TO THIS WORD

WAS THIS A TURN?

WAS CKCOR SCHEDULED?

CANCEL CKCOR

BRANCH ON RANGE
1- > 5 MILES
2- <= 5 MILES

IS TURN EVENLY DIVISIBLE BY
5 DEGREES?

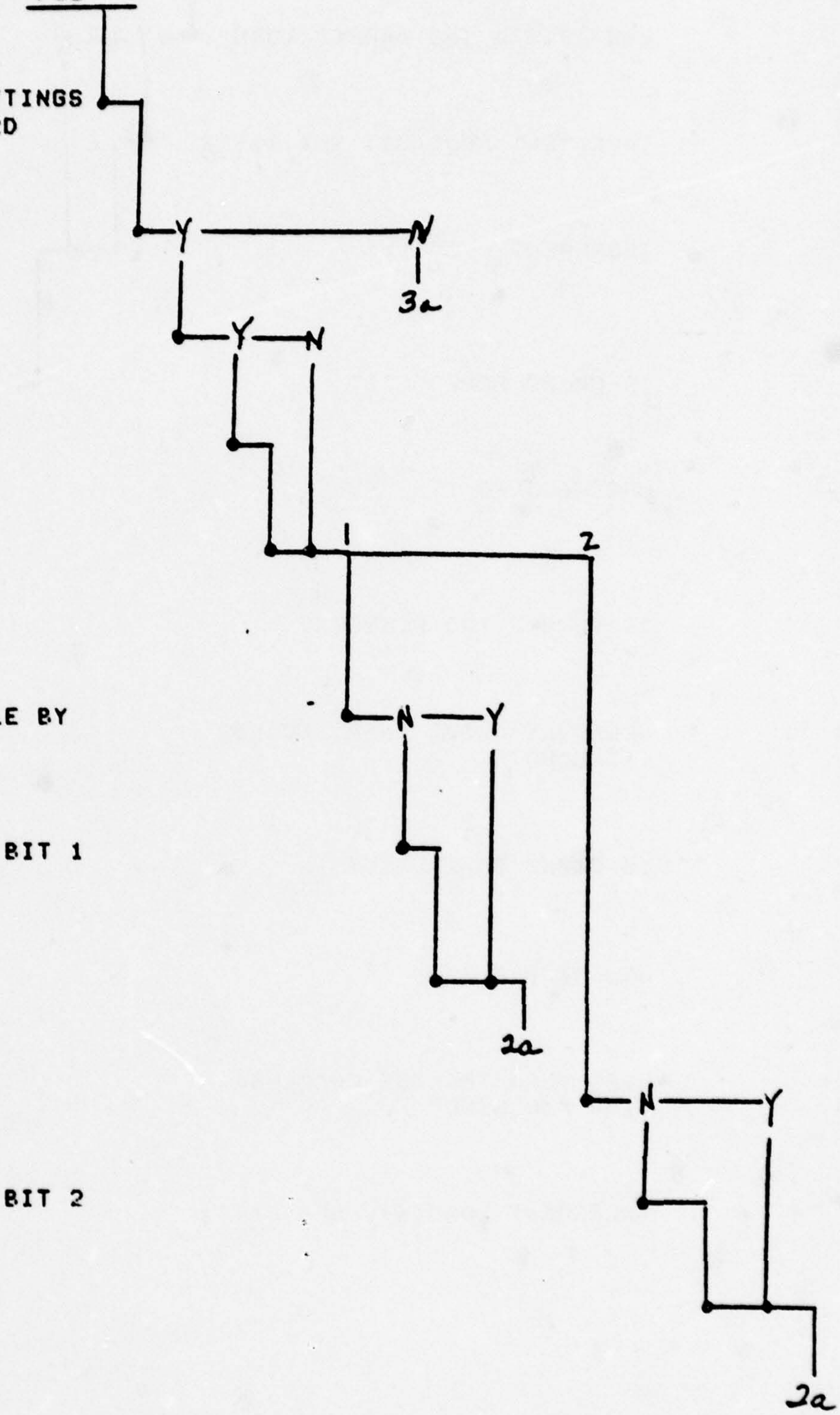
INCREMENT PVO5(1), SET BIT 1
IN PVO5(0)

INCREMENT PVO5(9)

IS TURN > 1 DEGREE?

INCREMENT PVO5(2), SET BIT 2

INCREMENT PVO5(10)



P05 FLOWCHART (SHEET 1 OF 3)

WAS THIS A 360 DEGREE TURN?

INCREMENT PV05(3), SET BIT 3

INCREMENT PV05(11)

IS CK120 SCHEDULED?

UNSCHEDULED IT

IS TURN > 120 DEGREES?

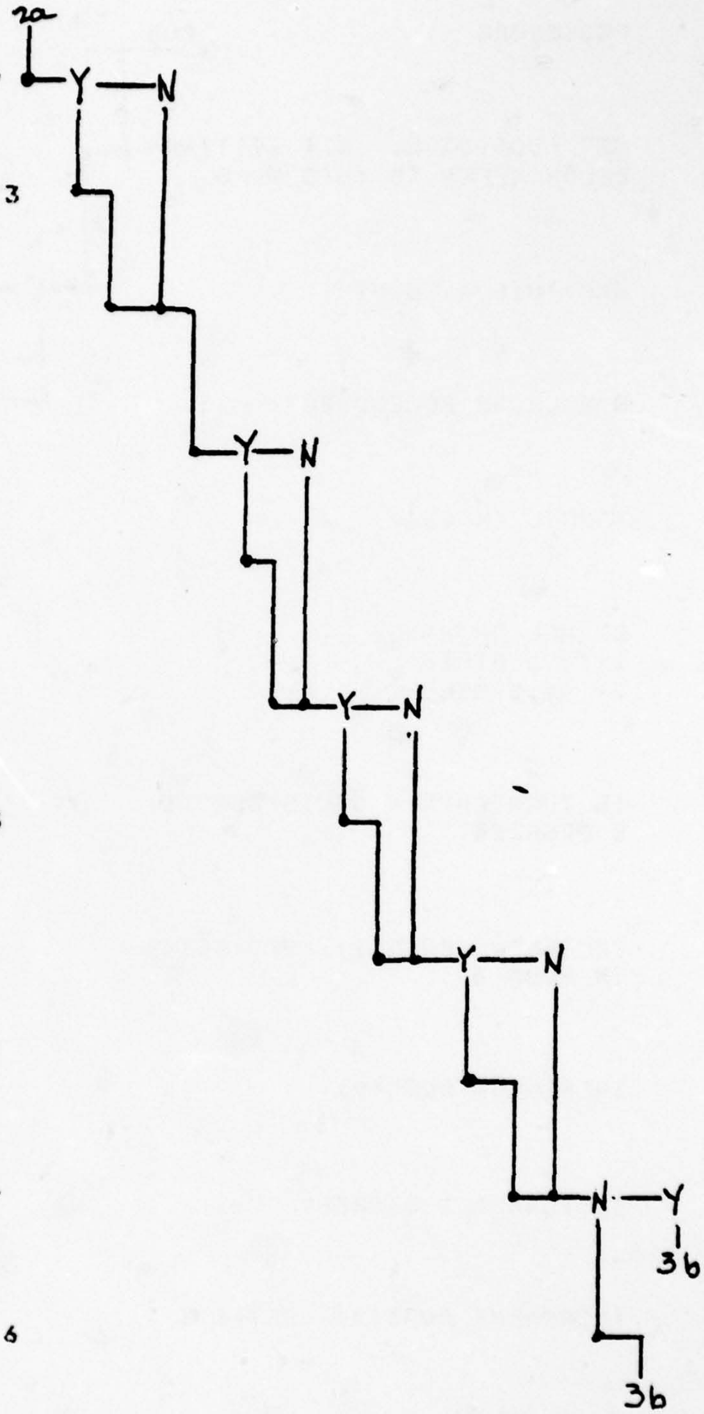
SCHEDULE CK120 IN 8 SECONDS
(TIMSCHD)

IS CKZN3 SCHEDULED?

UNSCHEDULE IT

DOES TURN INCLUDE COMPENSA-
TION FOR WIND?

INCREMENT PV05(6), SET BIT 6



P05 FLOWCHART (SHEET 2 OF 3)

IS THIS HEADING THE SAME AS
THE LAST ONE GIVEN?

INCREMENT PV05(7), SET BIT 7

INCREMENT PV05(14)

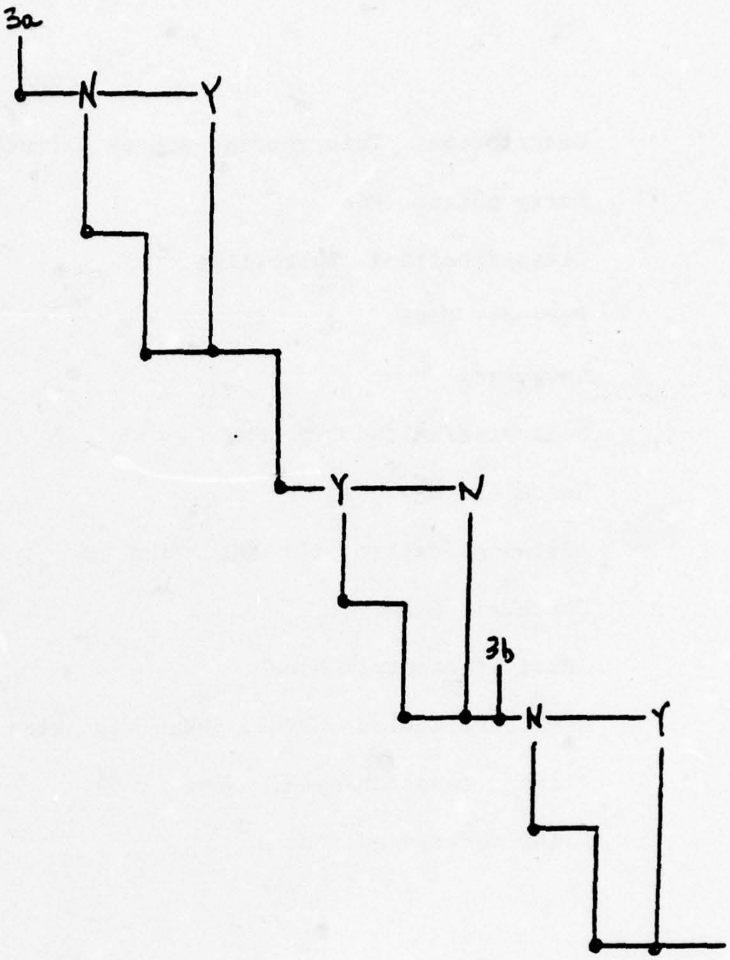
IS PV05 > 5?

INCREMENT PV05(8), SET BIT 8

ARE ALL BITS OF PV05(0) 0?

CALL EXPLAIN

RETURN



P05 FLOWCHART (SHEET 3 OF 3)

P06

Description: This routine scores azimuth position and trend messages.

Entry point: P06

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: TIMSCHD, CKCOR

Cancels: CKCOR

Input arguments: None

Common variables: PV06, SHUSH variables

Files created/changed: None

Files referenced: None

PROCEDURE P06

BRANCH ON MESSAGE TYPE
1-COURSE POSITION
2-COURSE TREND

WAS CALL CORRECT?

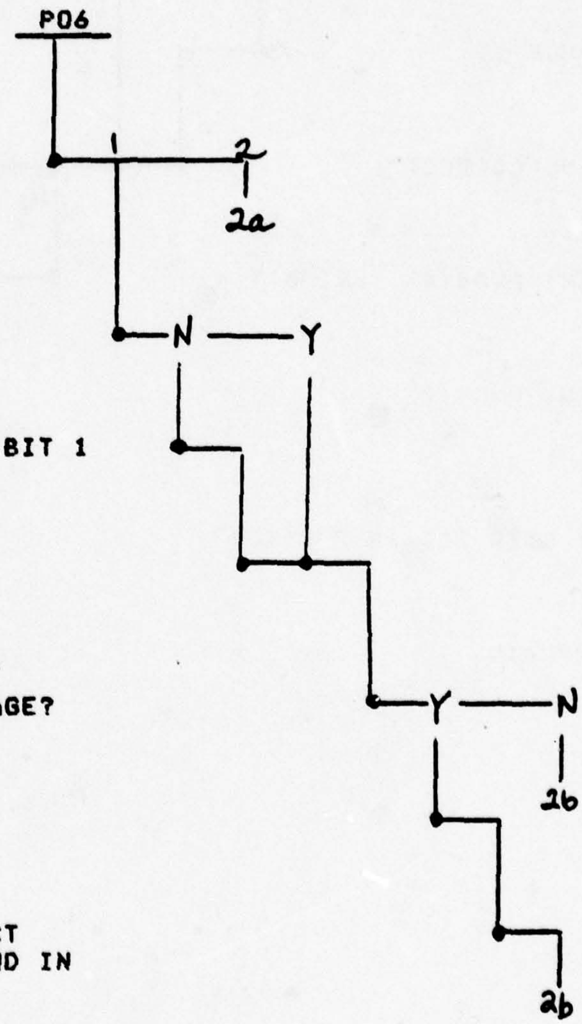
INCREMENT PVO6(1), SET BIT 1

INCREMENT PVO6(5)

WAS THIS A "WELL" MESSAGE?

INCREMENT PVO6(6)

SCHEDULE CKCOR TO DETECT
CORRECTIVE TURN ON TREND IN
3 SECONDS



P06 FLOWCHART (SHEET 1 OF 2)

IS CKCOR SCHEDULED?

UNSCHEDULE IT

WAS TREND CORRECT?

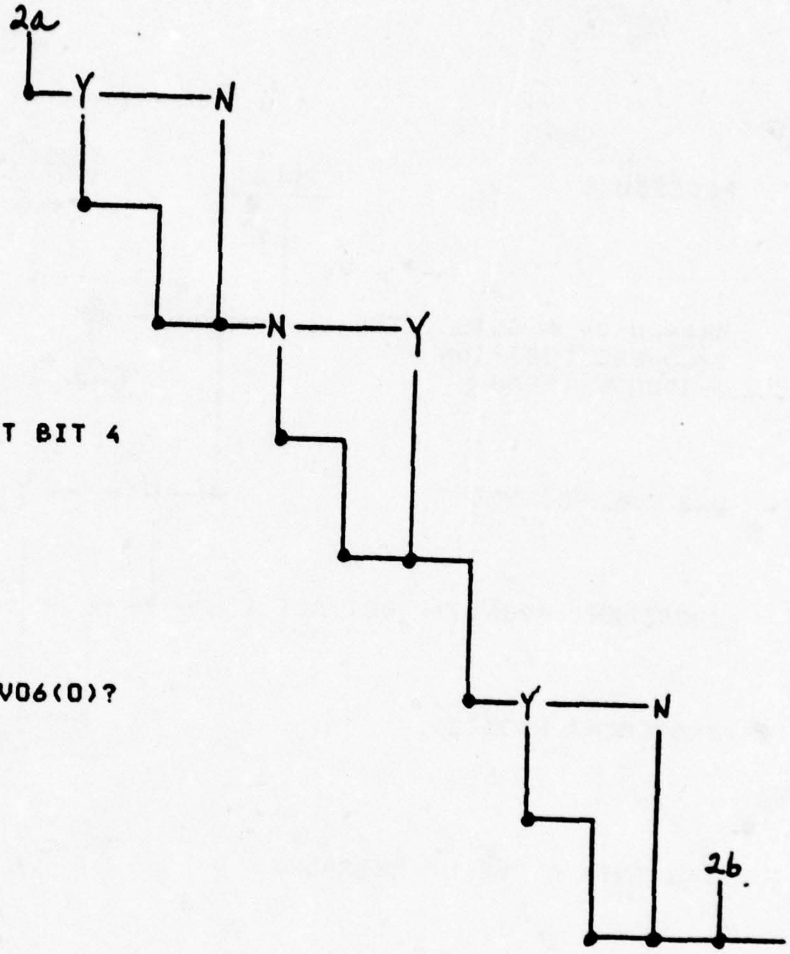
INCREMENT PV06(4), SET BIT 4

INCREMENT PV06(7)

ARE ANY BITS SET IN PV06(0)?

CALL EXPLAIN

RETURN



P06 FLOWCHART (SHEET 2 OF 2)

P07

Description: This routine scores glidepath position and trend messages.

Entry point: P07

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV07, SHUSH messages, PFGPP, PFGPT

Files created/changed: None

Files referenced: None

PROCEDURE

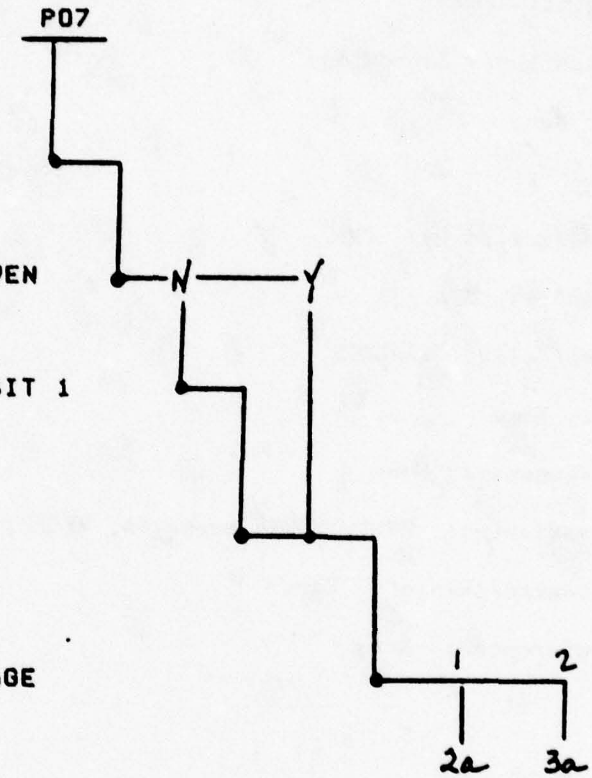
SET PV07(0) TO 0

WAS "BEGIN DESCENT" GIVEN

INCREMENT PV07(1) SET BIT 1
OF PV07(0)

INCREMENT PV07(5)

BRANCH ON TYPE OF MESSAGE
1-POSITION
2-TREND



P07 FLOWCHART (SHEET 1 OF 3)

WAS POSITION CORRECT?

INCREMENT PV07(2), SET BIT 2

INCREMENT PV07(6)

IS THIS POSITION DIFFERENT
FROM LAST?

WAS A TREND MESSAGE GIVEN?

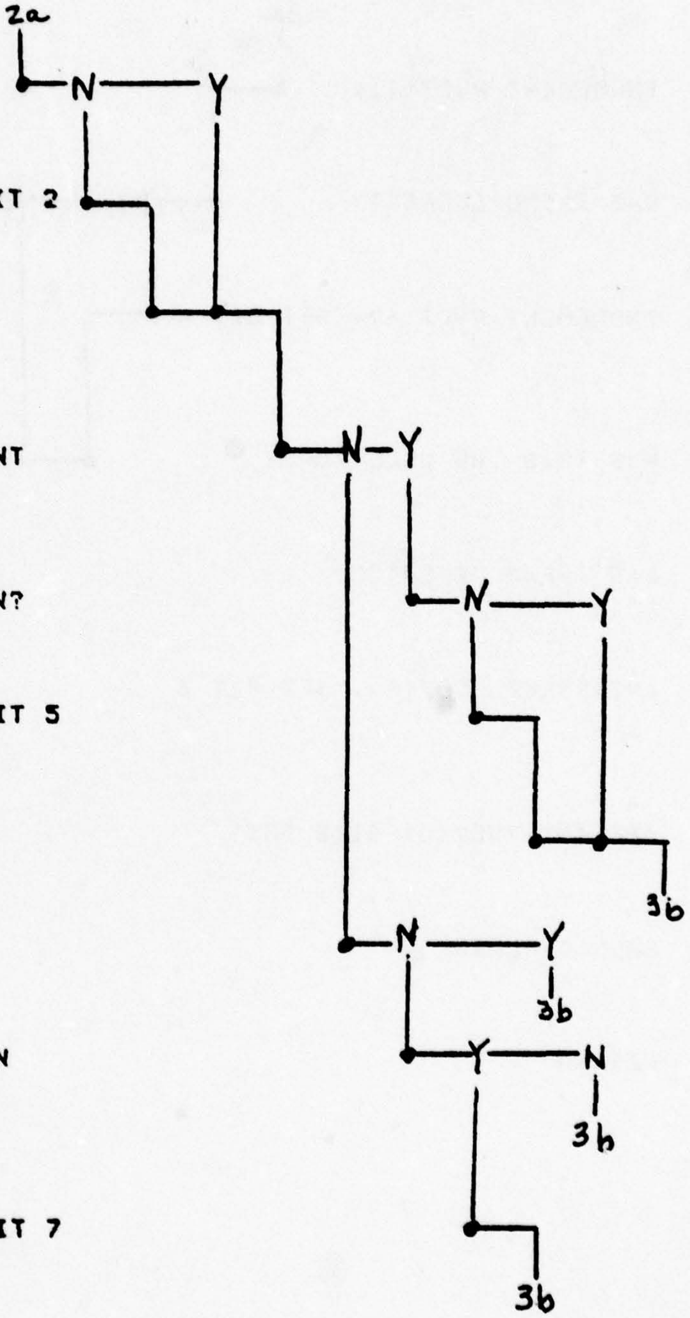
INCREMENT PV07(5), SET BIT 5

INCREMENT PV07(11)

WAS THIS THE WELL ZONE?

WAS A TREND GIVEN BETWEEN
THESE IDENTICAL POSITION
MESSAGES

INCREMENT PV07(7), SET BIT 7



P07 FLOWCHART (SHEET 2 OF 3)

INCREMENT PV07(11)

WAS TREND CORRECT?

INCREMENT PV07(4), SET BIT 4

WAS THIS THE WELL ZONE?

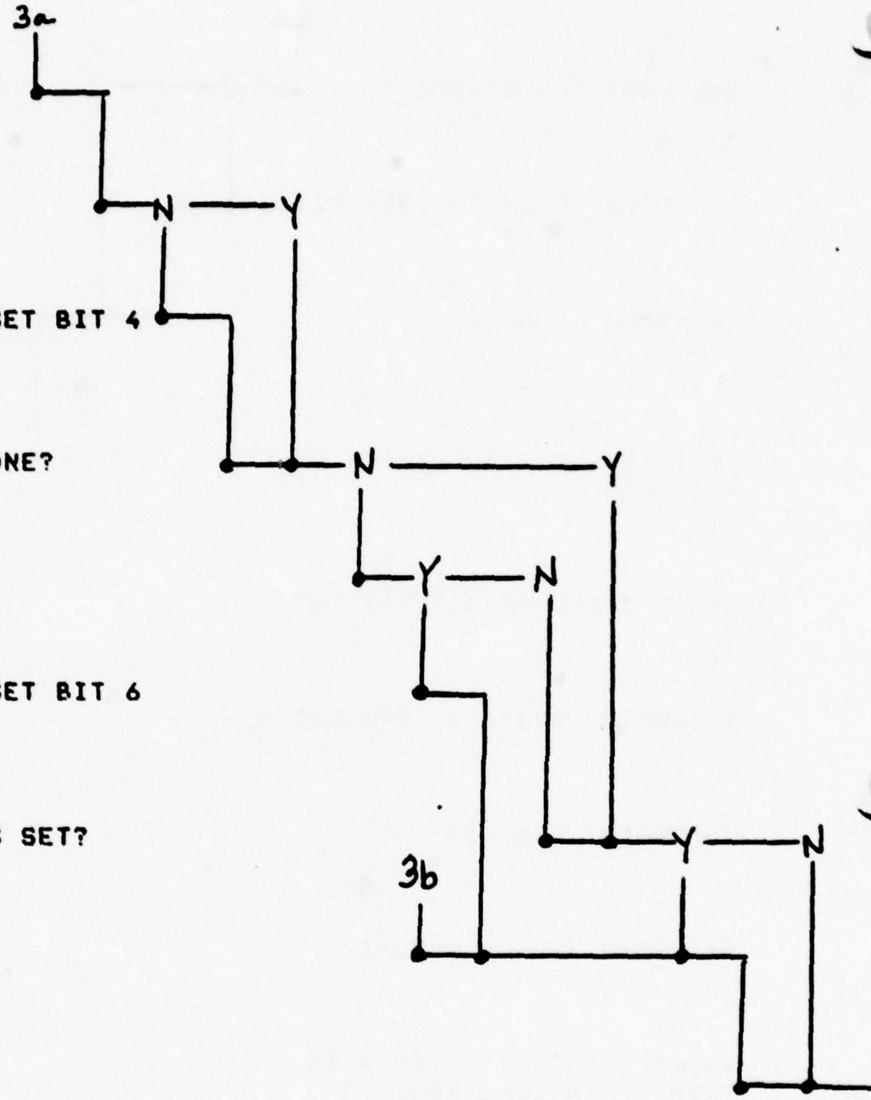
WAS TREND REPEATED?

INCREMENT PV07(6), SET BIT 6

ARE ANY PV07(0) BITS SET?

CALL EXPLAIN

RETURN



PO7 FLOWCHART (SHEET 3 OF 3)

P08

Description: This routine scores mile marks for accuracy.

Entry point: P08

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: PCHK, EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV08, ACRNG, CTRNG

Files created/changed: None

Files referenced: None

PROCEDURE

P08

CLEAR PV08(0)

INCREMENT PV08(5)

CLEAR BIT IN PV08(6) CORRES-
PONDING TO THIS MILE MARK

WAS MILE MARK GIVEN AT THE
RIGHT TIME?

INCREMENT PV08(2) SET BIT 12
OF PV08(0)

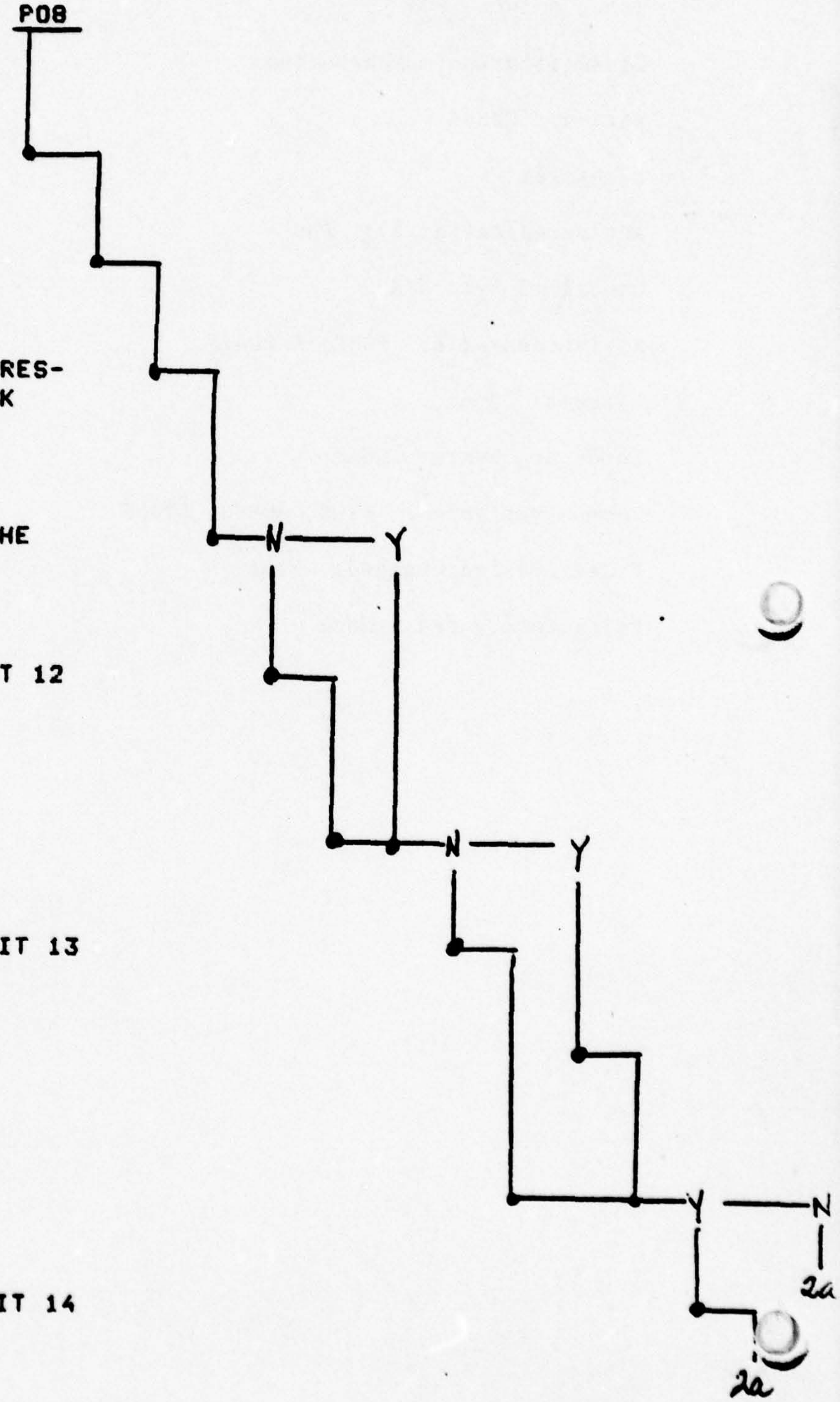
WAS MILE MARK CORRECT?

INCREMENT PV08(3), SET BIT 13

REMOVE IT FROM QUEUE

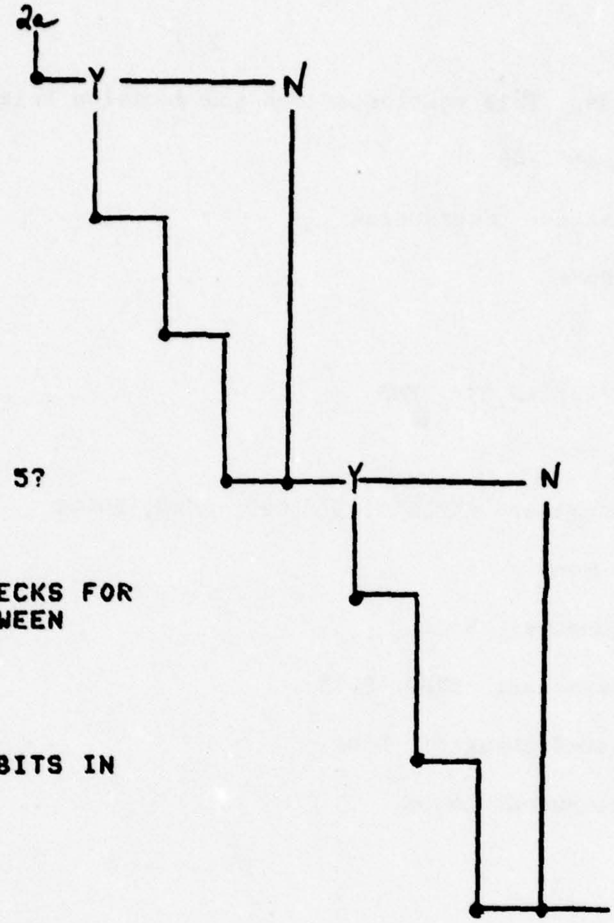
WAS IT REPEATED?

INCREMENT PV08(4), SET BIT 14



P08 FLOWCHART (SHEET 1 OF 2)

WERE ANY BITS SET IN FV08(0)?



SET MILE NUMBER BIT

CALL EXPLAIN

WAS THIS MILE MARK > 5?

SCHEDULE OMISSION CHECKS FOR OTHER MILE MARKS BETWEEN THIS ONE AND 5 MILES

SET OTHER MILE MARK BITS IN FV08(6)

RETURN

P08 FLOWCHART (SHEET 2 OF 2)

P09

Description: This routine scores the decision height message.

Entry point: P09

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, TIMSCHD, CKWO, RDACT

Cancels: None

Input arguments: None

Common variables: PV09, PV15

Files created/changed: None

Files referenced: None

PROCEDURE

IS BIT 1 OF PV09 CLEAR, IE.
WAS MESSAGE GIVEN BEFORE?

SET BIT 9

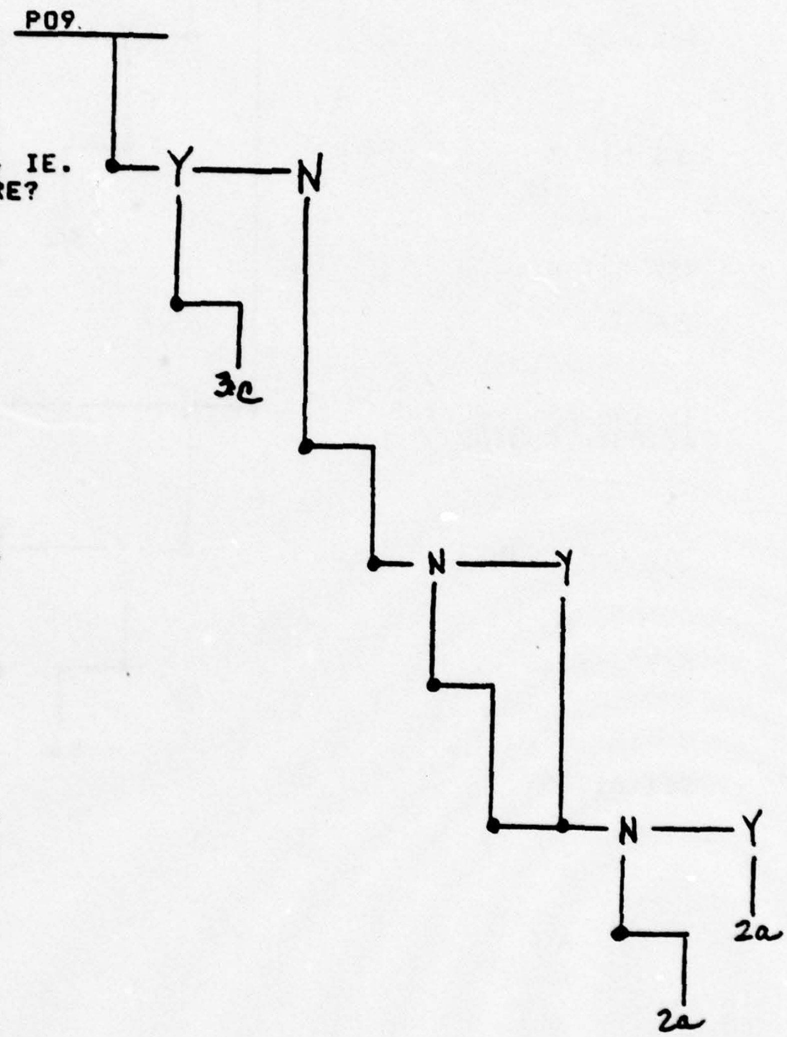
CLEAR BIT 1

IS RANGE < = .85 MILES?

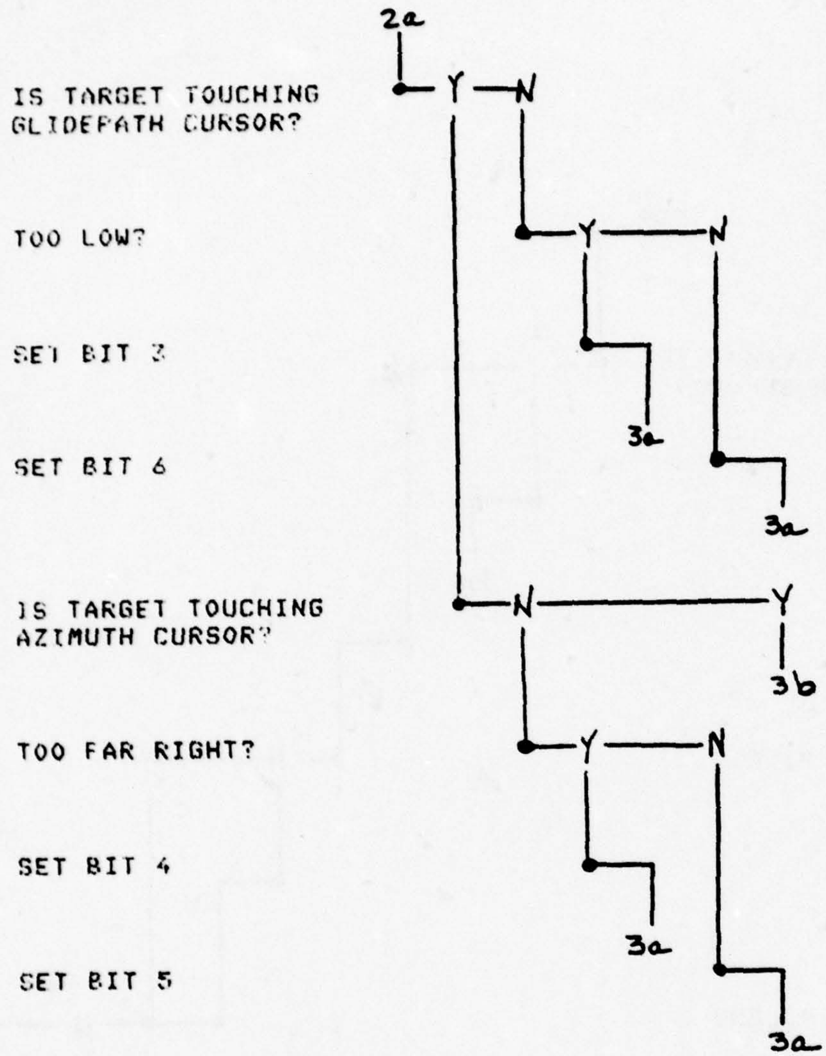
SET BIT 7

IS RANGE > = .65 MILES?

SET BIT 8



P09 FLOWCHART (SHEET 1 OF 3)



P09 FLOWCHART (SHEET 2 OF 3)

WAS NEXT ADVISORY A
"TOO" MESSAGE?

SET BIT 2

WAS POSITION CORRECT?

CLEAR BITS 3-6

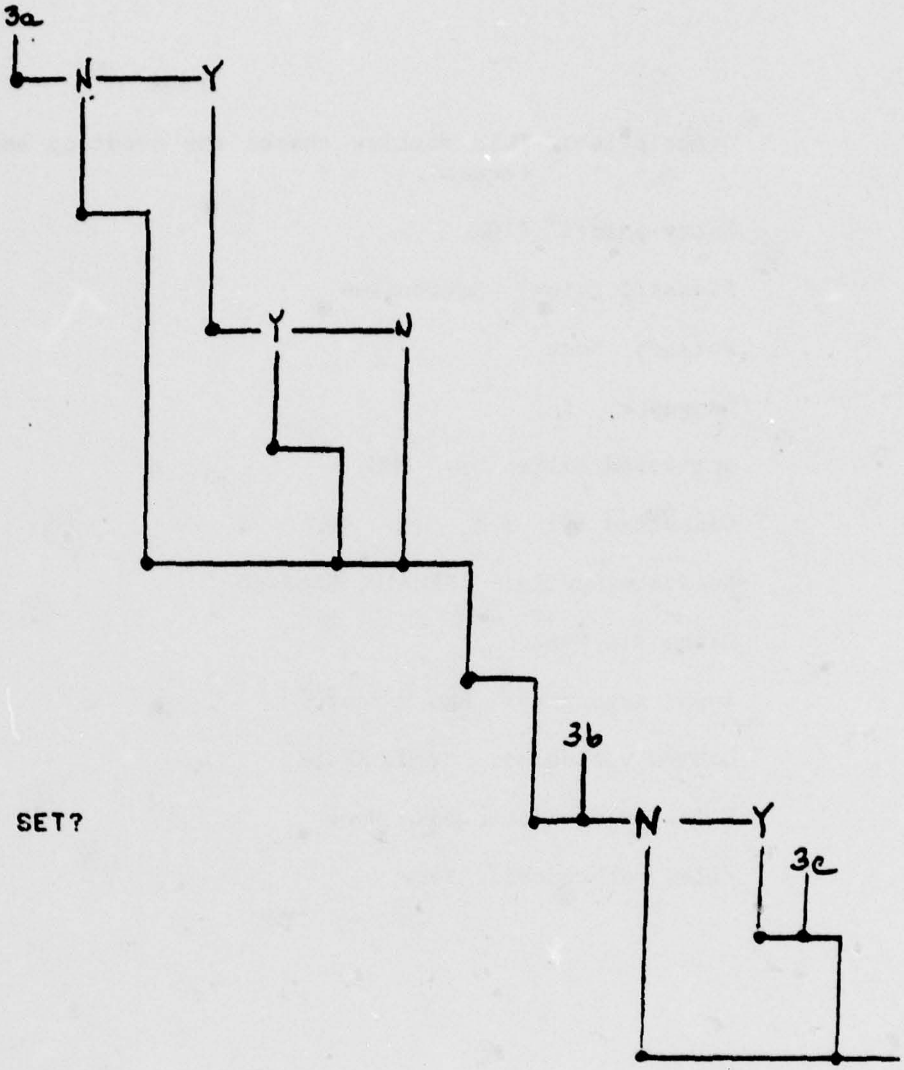
SET BIT 4 OF PV15

SCHEDULE CKWO IN
5 SECONDS

WERE ANY BITS OF PV09 SET?

CALL EXPLAIN

RETURN



P09 FLOWCHART (SHEET 3 OF 3)

P10A

Description: This routine checks the accuracy and timing of the clearance request.

Entry point: P10A

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, RNGSCHD

Cancels: None

Input arguments: RNG — range

Common variables: PV10, KYCLR

Files created/changed: None

Files referenced: None

PROCEDURE P10A

IS THIS THE FIRST CLEARANCE REQUEST?

CLEAR BIT 14 OF PV10

WILL TOWER GIVE CONTINUE?

SET BIT 15 OF PV10

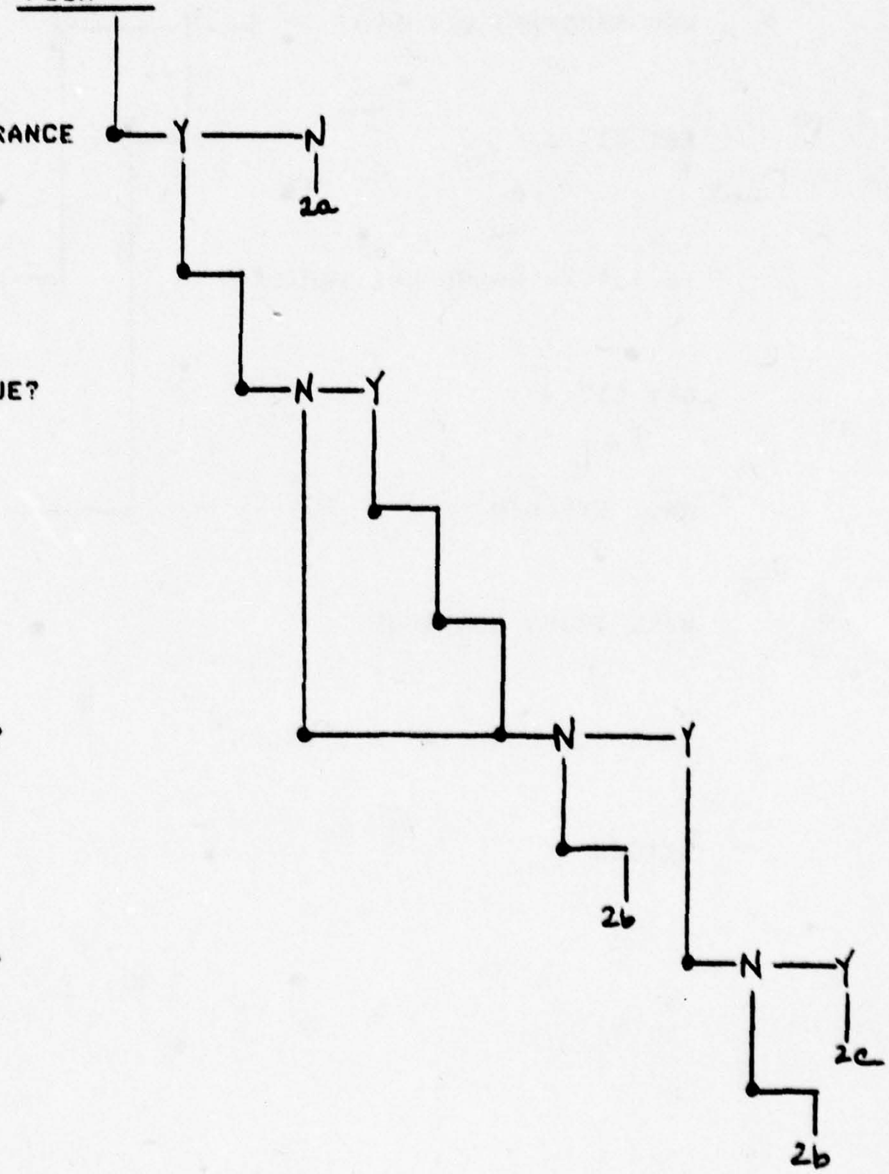
SCHEDULE CKCLR

IS RANGE < = 3.1 MILES?

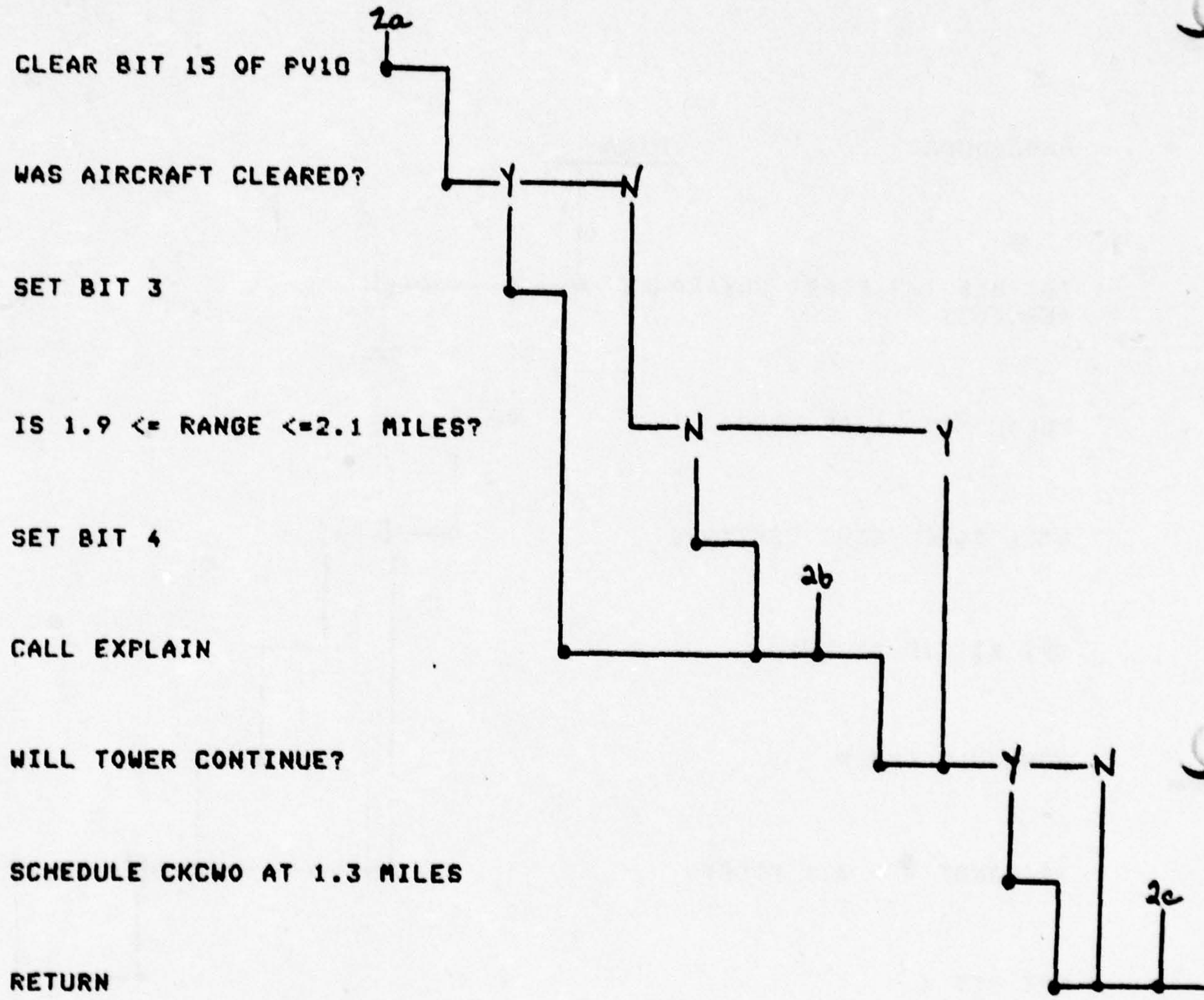
SET BIT 1

IS RANGE > = 2.9 MILES?

SET BIT 2



P10A FLOWCHART (SHEET 1 OF 2)



P10A FLOWCHART (SHEET 2 OF 2)

P10B

Description: P10B scores the wind message.

Entry point: P10B

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: SHUSH variables, PV10

Files created/changed: None

Files referenced: None

PROCEDURE

WAS AIRCRAFT CLEARED?
(KYCLR)

SET BIT 6 OF PV10

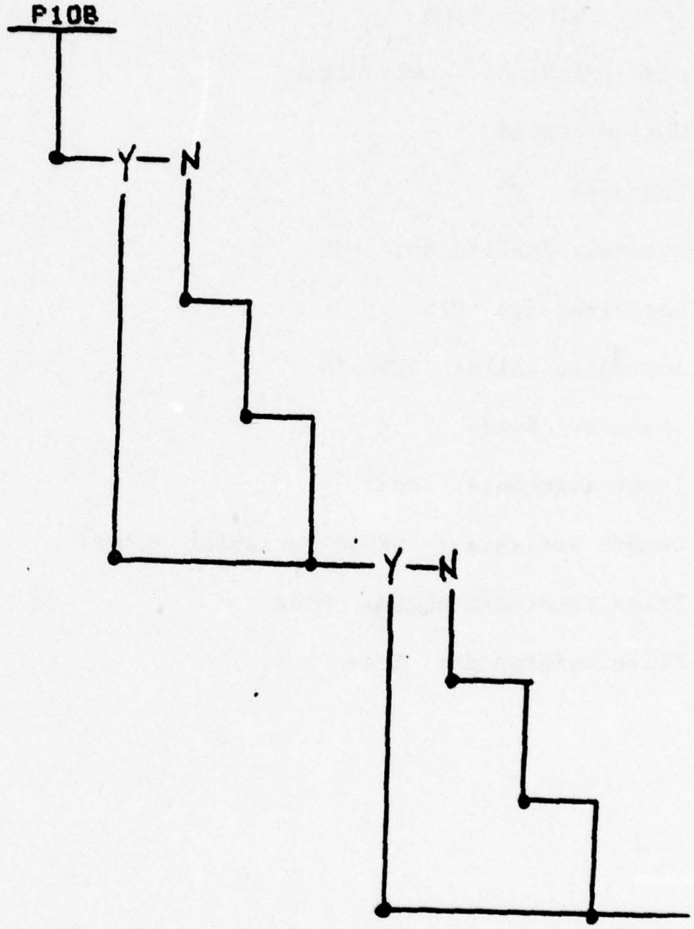
CALL EXPLAIN

WAS WIND CORRECT?

SET BIT 5 OF PV10

CALL EXPLAIN

RETURN



P10B FLOWCHART (SHEET 1 OF 1)

P10C

Description: This routine scores the clearance message.

Entry point: P10C

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: RNG — aircraft range

Common variables: SHUSH variables, PV10

Files created/changed: None

Files referenced: None

PROCEDURE P1OC

WAS AIRCRAFT CLEARED?

SET BIT 7 OF PV1O

WAS WIND GIVEN?

SET BIT 8 OF PV1O

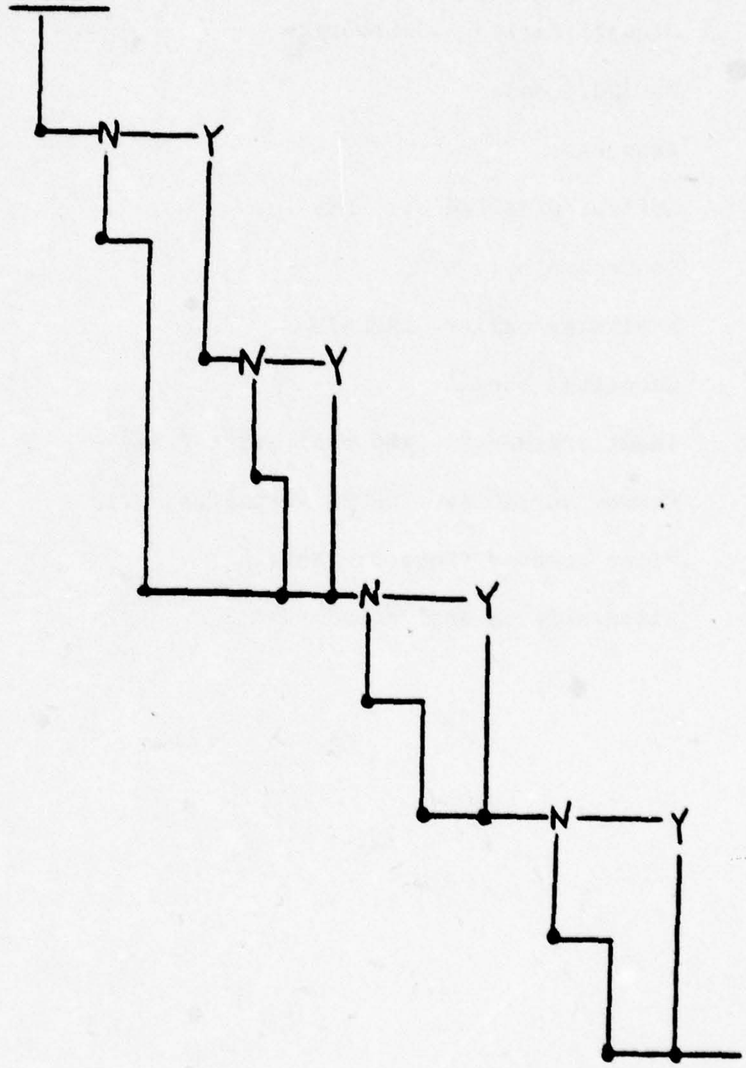
IS RANGE > = 1 MILE?

SET BIT 9 OF PV1O

ARE ALL BITS 7-9 O?

CALL EXPLAIN

RETURN



P1OC FLOWCHART (SHEET 1 OF 1)

P10D

Description: This routine checks the accuracy of the clearance waveoffs.

Entry point: P10D

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

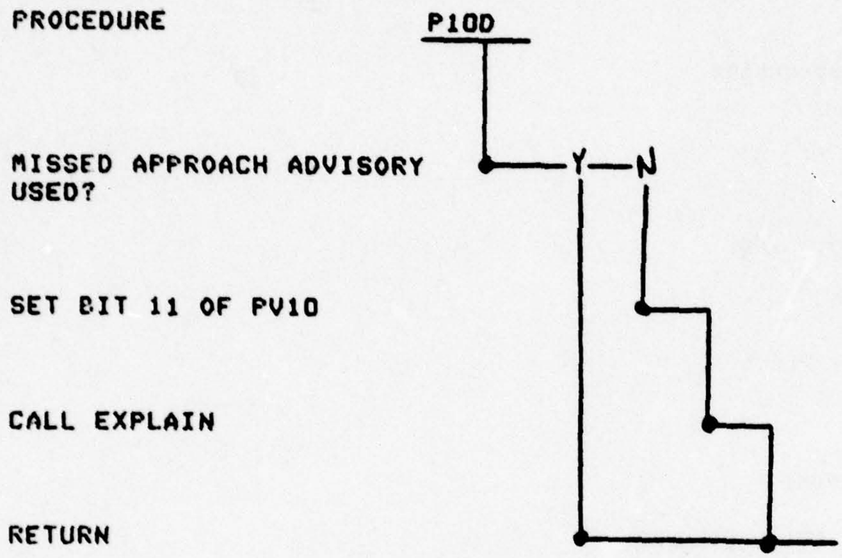
Cancels: None

Input arguments: None

Common variables: PV10, CTEMERG

Files created/changed: None

Files referenced: None



P10D FLOWCHART (SHEET 1 OF 1)

P11

Description: P11 scores the over landing threshold advisory.

Entry point: P11

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, RDACT

Cancels: None

Input arguments: None

Common variables: PV11, CTRNG

Files created/changed: None

Files referenced: None

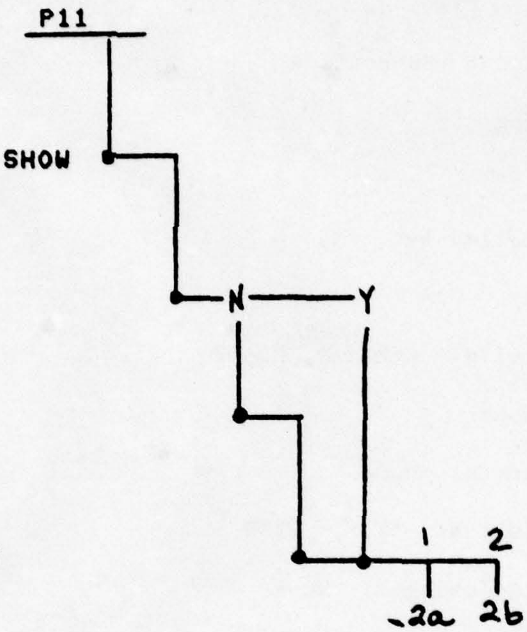
PROCEDURE

CLEAR BIT 1 OF PV11 TO SHOW
ADVISORY GIVEN

WAS TIMING CORRECT?

SET BIT 2 OF PV11

BRANCH ON NEXT ADVISORY
1-COURSE POSITION
2-OVER



P11 FLOWCHART (SHEET 1 OF 2)

WAS COURSE POSITION CORRECT?

SET BIT 4 OF PV11

WAS NEXT ADVISORY OVER?

SET BIT 5 OF PV11

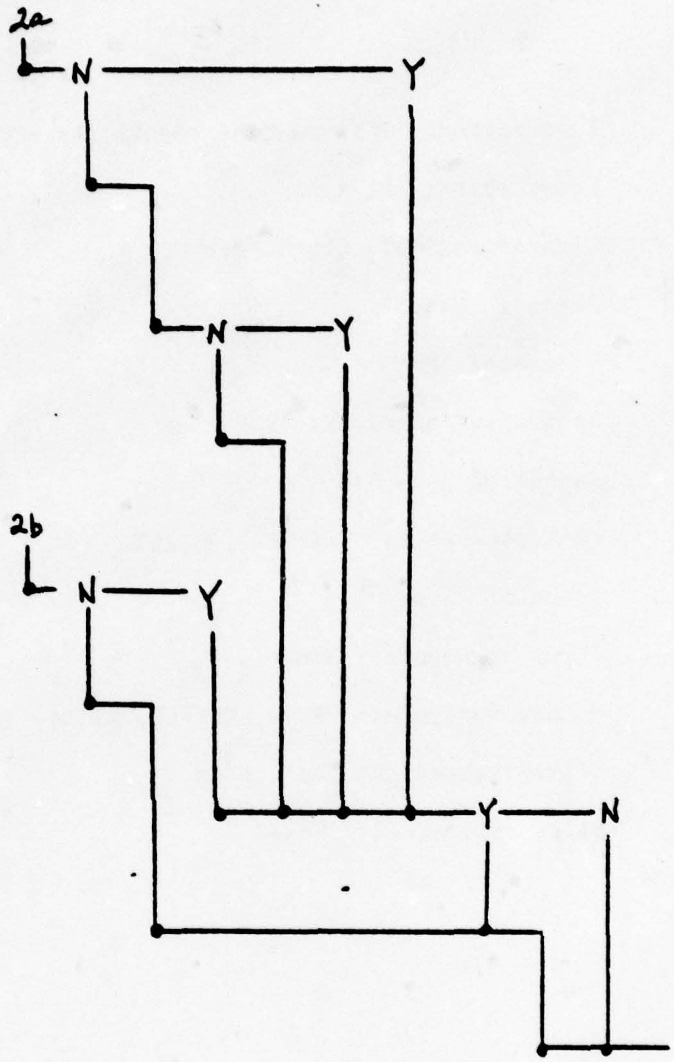
WAS AIRCRAFT ON COURSE?

SET BIT 4 OF PV11

WERE ANY BITS OF PV11 SET?

CALL EXPLAIN

RETURN



P11 FLOWCHART (SHEET 2 OF 2)

P12A

Description: This routine checks the accuracy of the rollout instructions.

Entry point: P12A

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, RDACT

Cancels: None

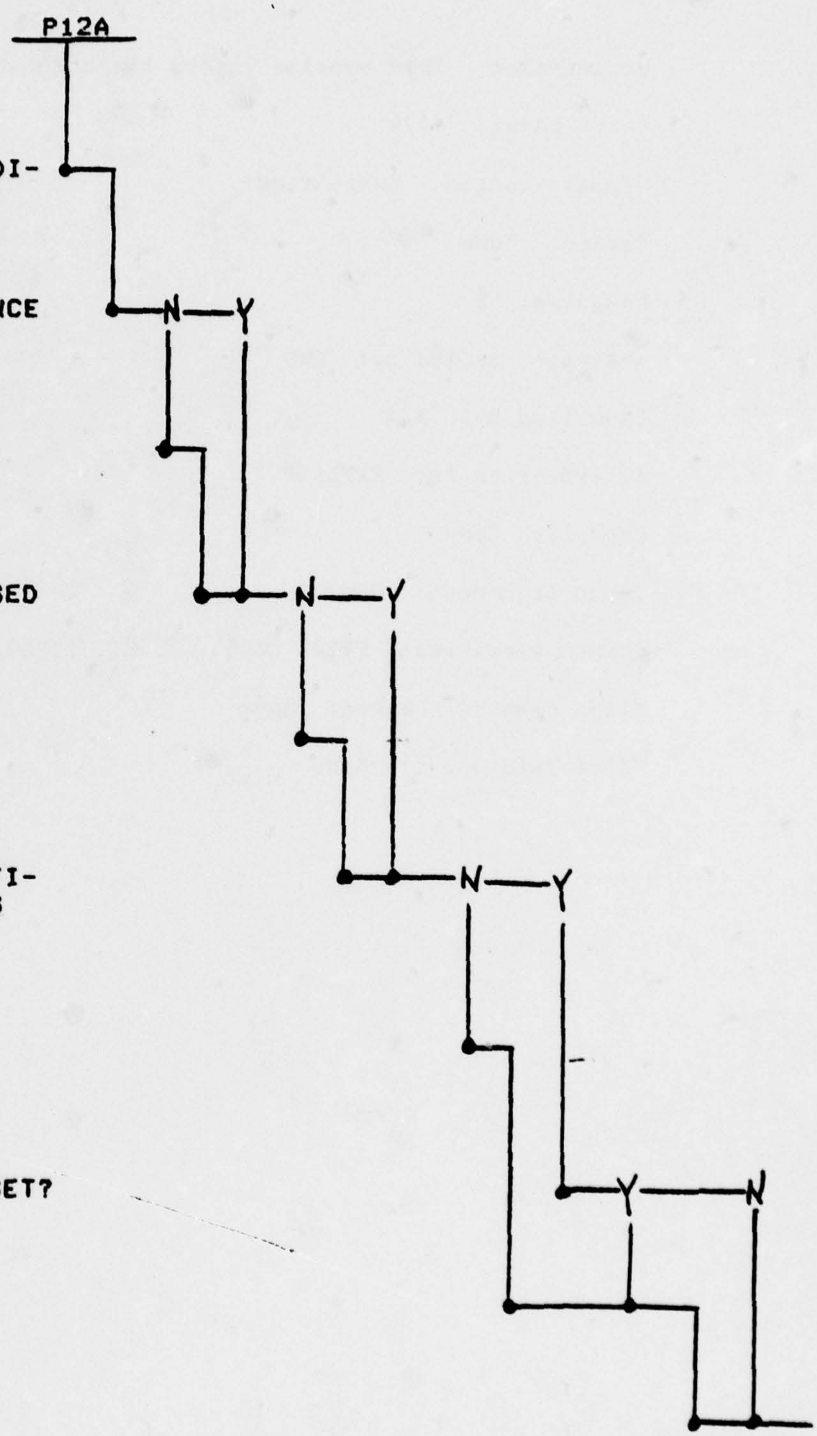
Input arguments: None

Common variables: PV12, KYFREQ, KYIC4

Files created/changed: None

Files referenced: None

PROCEDURE



CLEAR BIT 1 OF PV12 TO INDICATE ROLLOUT WAS GIVEN

HAS 20 SECONDS ELAPSED SINCE "OVER" WAS ISSUED?

SET BIT 2 OF PV12

WAS RADIO FREQUENCY RELEASED WITHIN 10 SECONDS?

SET BIT 3

WAS PATTERN CONTROLLER NOTIFIED? (MESSAGE GIVEN, ICS SELECTED)

SET BIT 4

ARE ANY BITS 2-4 OF PV11 SET?

CALL EXPLAIN

RETURN

P12A FLOWCHART (SHEET 1 OF 1)

P12B

Description: This routine checks the accuracy of the handoff.

Entry point: P12B

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV12, ACCS, CTFREQ, ACRNG, KYIC4

Files created/changed: None

Files referenced: None

PROCEDURE P12B

CLEAR BIT 5 OF PV12 TO SHOW
HANDOFF WAS GIVEN

WAS CALL SIGN CORRECT?

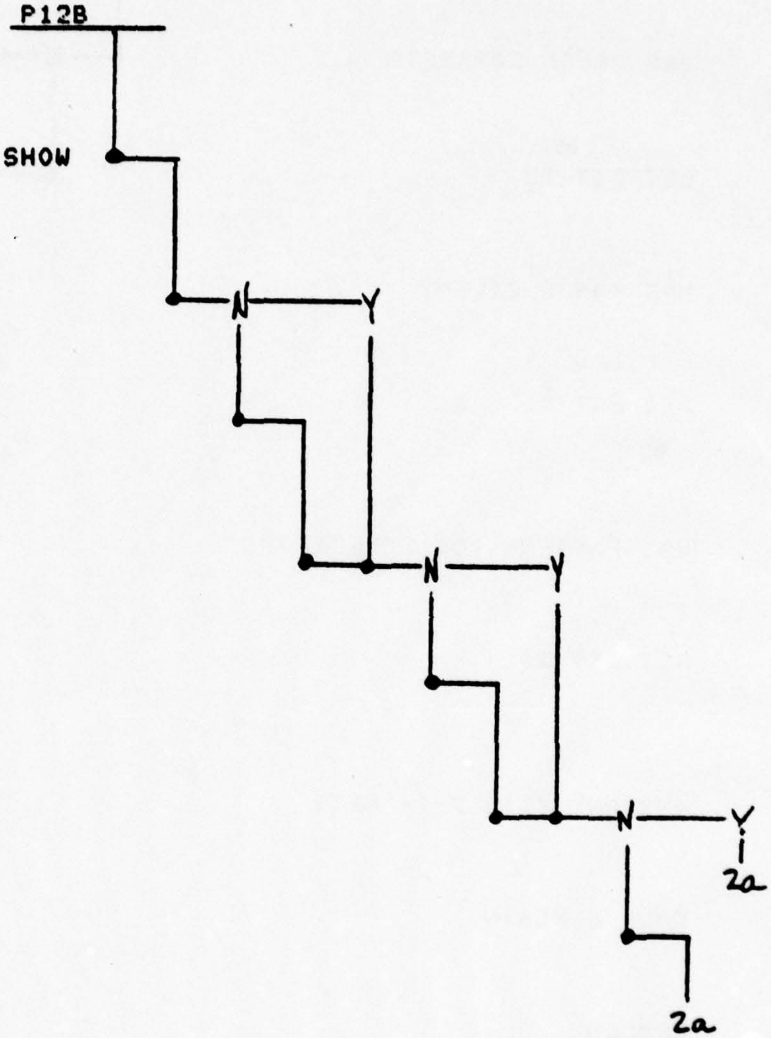
SET BIT 7

WAS FREQUENCY CORRECT?

SET BIT 8

WAS BUTTON CORRECT?

SET BIT 9



P12B FLOWCHART (SHEET 1 OF 2)

IS THIS A MISSED APPROACH?

WAS RANGE CORRECT?

SET BIT 10

WAS RANGE GIVEN?

SET BIT 11

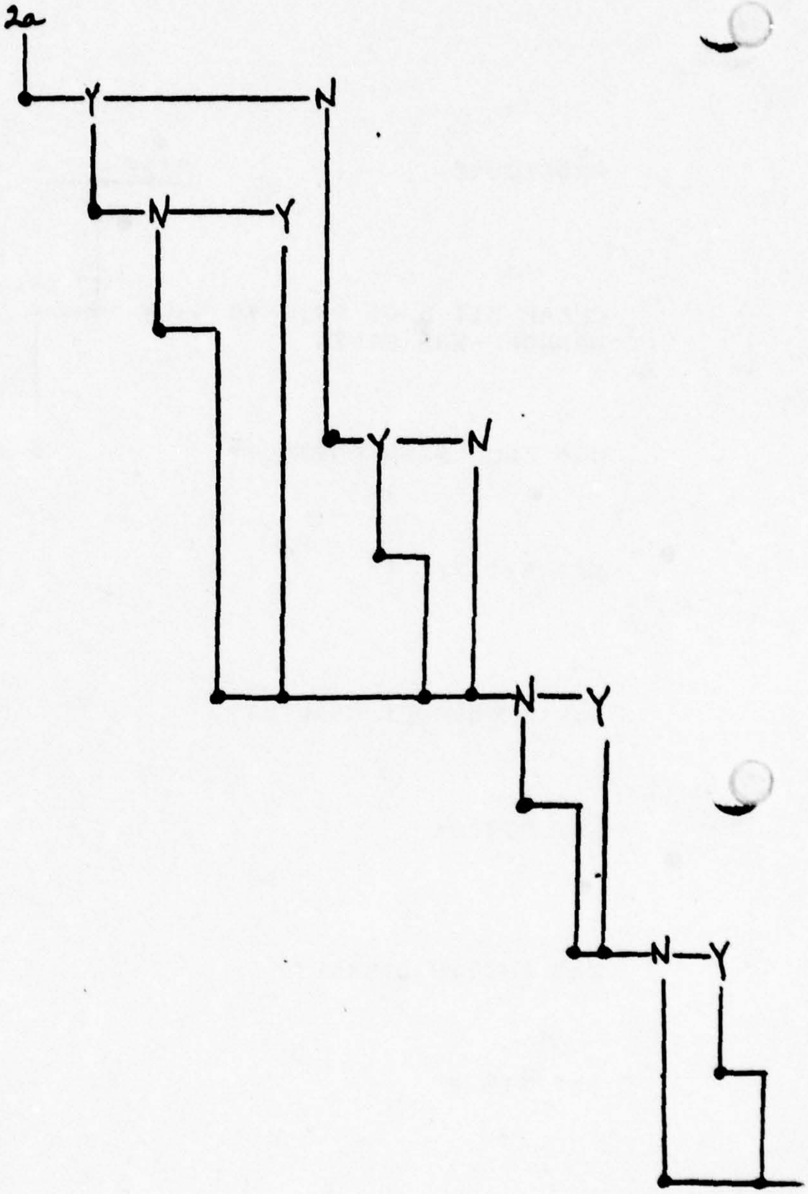
WAS PATTERN ICS SELECTED?

SET BIT 14

ANY ANY BITS 7-14 SET?

CALL EXPLAIN

RETURN



P12B FLOWCHART (SHEET 2 OF 2)

P12C

Description: This routine checks for proper termination of the handoff procedure.

Entry point: P12C

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN, RDACT

Cancels: None

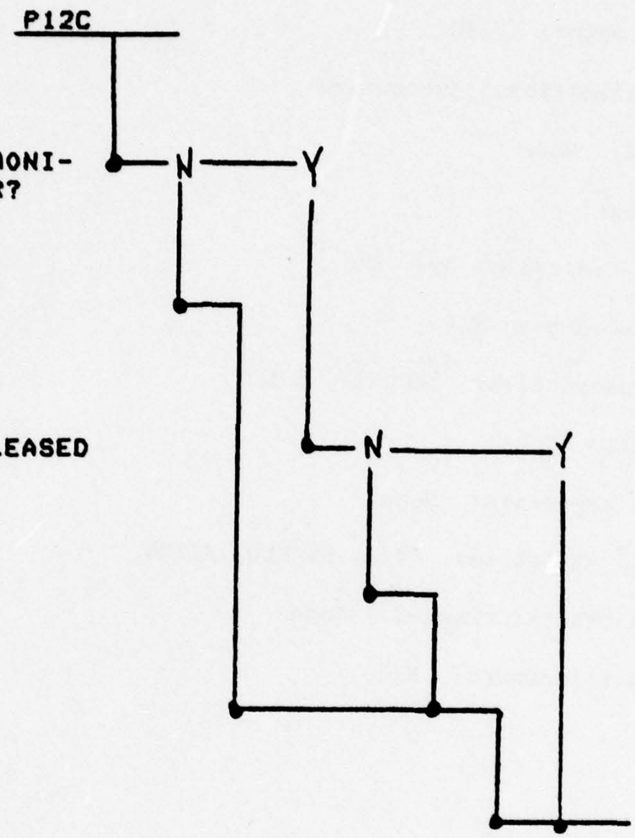
Input arguments: None

Common variables: PV12, KYFREQ, KYIC4

Files created/changed: None

Files referenced: None

PROCEDURE



WAS FREQUENCY AND ICS MONITORED THROUGH C/S RADAR?

SET BIT 12 OF PV12

WAS RADIO FREQUENCY RELEASED WITHIN 5 SECONDS?

SET BIT 13 OF PV12

CALL EXPLAIN

RETURN

P12C FLOWCHART (SHEET 1 OF 1)

P15A — P15D

Description: This set of routines scores waveoffs other than those due to clearance problems.

Entry point: P15A, P15B, P15C, P15D

Classification: Subroutines

Period: None

Language: F

Activated/called by: EXPLAIN

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: PV15, SHUSH Messages

Files created/changed: None

Files referenced: None

PROCEDURE

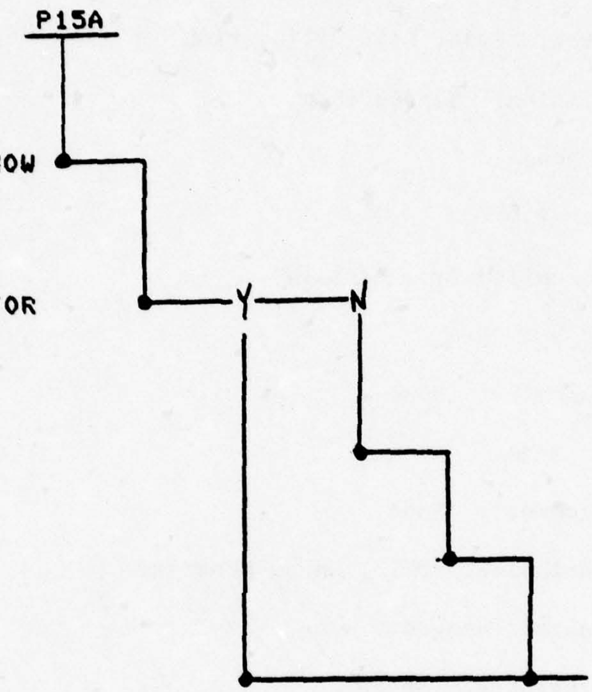
CLEAR PV15 BIT 1 TO SHOW
WAVEOFF WAS GIVEN

WAS CORRECT R/T USED FOR
TYPE OF APPROACH?

SET BIT 6

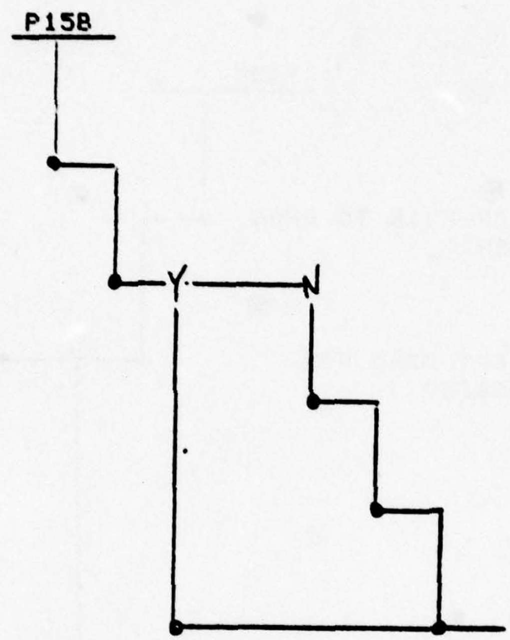
CALL EXPLAIN

RETURN



P15A FLOWCHART (SHEET 1 OF 1)

PROCEDURE



CLEAR BIT 2 OF PV15

WAS CORRECT R/T USED?

SET BIT 6

CALL EXPLAIN

RETURN

P15B FLOWCHART (SHEET 1 OF 1)

PROCEDURE

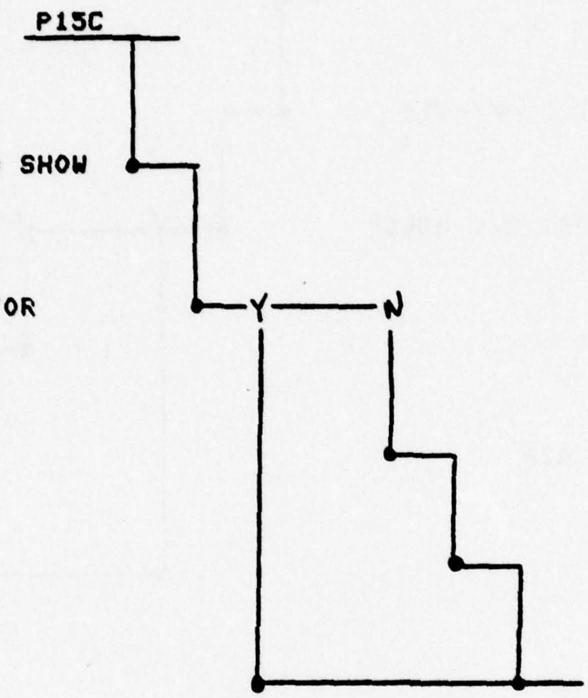
CLEAR BIT 3 OF PV15 TO SHOW
WAVE OFF GIVEN

WAS CORRECT R/T USED FOR
TYPE OF APPROACH?

SET BIT 6

CALL EXPLAIN

RETURN



P15C FLOWCHART (SHEET 1 OF 1)

PROCEDURE P15D

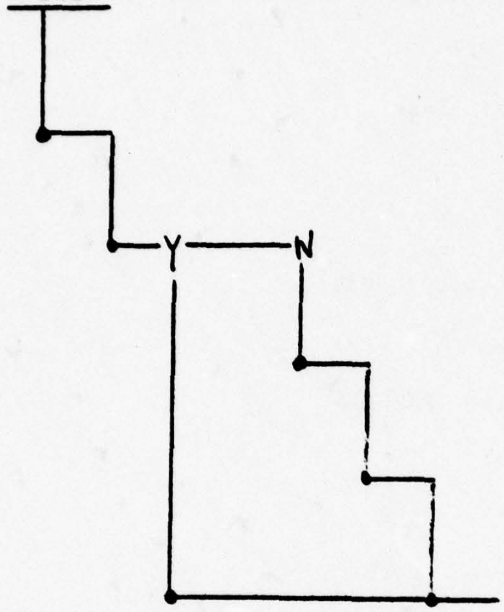
CLEAR BIT 4 OF PV15

WAS CORRECT R/T USED?

SET BIT 6

CALL EXPLAIN

RETURN



P15D FLOWCHART (SHEET 1 OF 1)

P17

Description: This routine ensures that the mike is unkeyed after "over" messages.

Entry point: P17

Classification: Task

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV17

Files created/changed: None

Files referenced: None

PROCEDURE P17

3 SECONDS HAS ELAPSED SINCE "OVER" WAS GIVEN

INCREMENT PV17 (3)

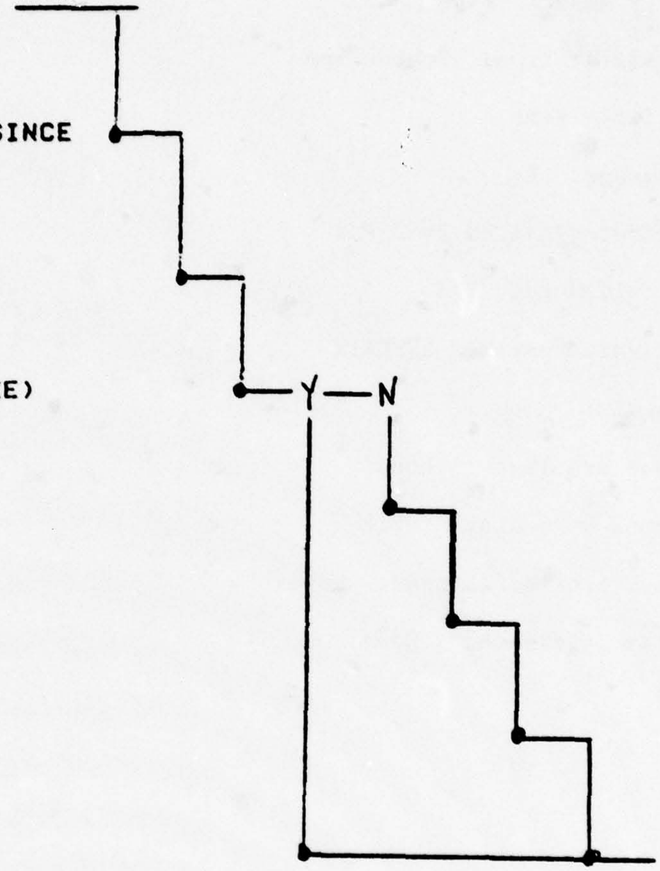
IS MIKE UNKEYED? (KYMIKE)

SET BIT 1 OF P17 (0)

INCREMENT PV17 (1)

CALL EXPLAIN

DIE



P17 FLOWCHART (SHEET 1 OF 1)

P18A

Description: This routine detects the too rapid issuance of advisories.

Entry point: P18A

Classification: Subroutine

Period: None

Language: F

Activated/called by: PMS

Cancelled by: N/A

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV18

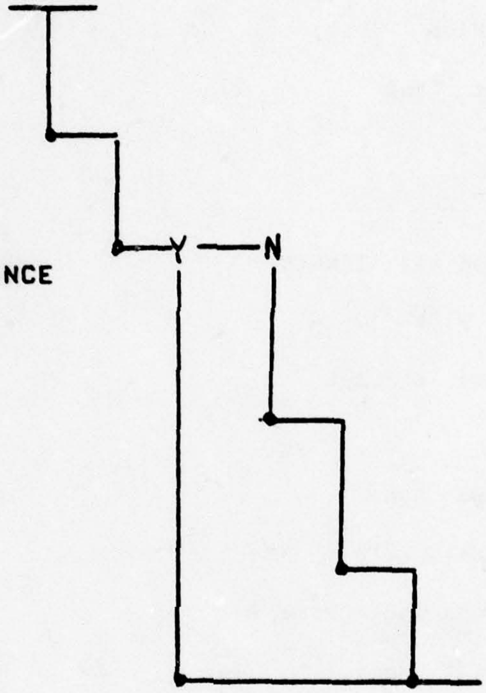
Files created/changed: None

Files referenced: None

PROCEDURE P18A

INCREMENT PV18(4)

STUDENT VOICE ACTIVE
HAS IT BEEN 1 SECOND SINCE
END OF LAST ADVISORY?



INCREMENT PV18(1), SET
PV18(0) TO 1B1

CALL EXPLAIN

RETURN

P18A FLOWCHART (SHEET 1 OF 1)

P18B

Description: P18B detects pauses in controller advisories of more than five seconds.

Entry point: P18B

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL

Cancelled by: P18B

Activates/calls: EXPLAIN

Cancels: None

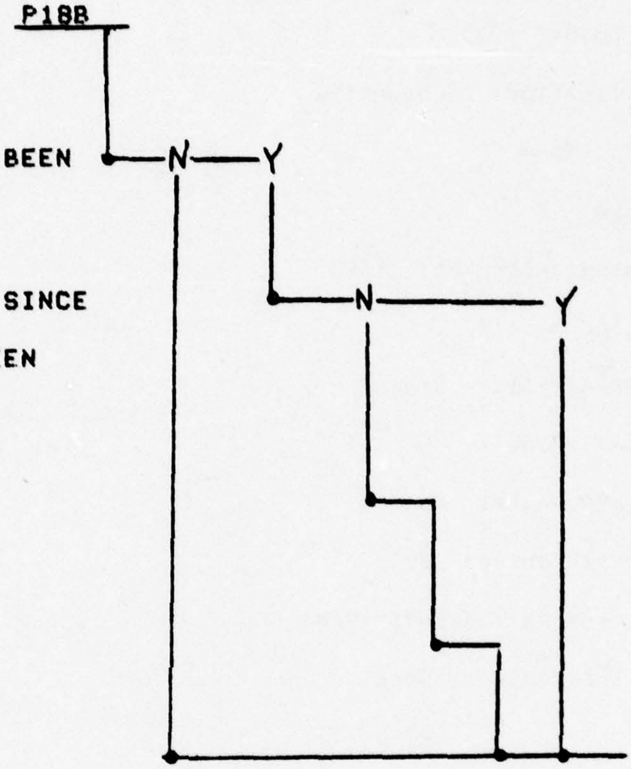
Input arguments: None

Common variables: PV18

Files created/changed: None

Files referenced: None

PROCEDURE P18B



HAS DO NOT ACKNOWLEDGE BEEN GIVEN?

5 SECONDS HAVE ELAPSED SINCE END OF SPEECH
HAS ANOTHER ADVISORY BEEN STARTED?

INCREMENT PV18(2), SET PV18(0) TO 1B2

CALL EXPLAIN

DIE

P18B FLOWCHART (SHEET 1 OF 1)

P19

Description: This routine scores alignment checking procedures.

Entry point: P19

Classification: Subroutine

Period: None

Language: F

Activated/called by: TZEC

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: PV19

Files created/changed: None

Files referenced: None

Performance Measurement Supporting Routines

Data collection for purposes of analyzing the performance of the trainee is accomplished by these routines. In general, they detect events in the environment and set indicators as appropriate.

RDACT

Description: This subroutine reads through the activity file buffer, setting conditions as specified, until it encounters a record of the specified type. During phase 2, it reads this buffer information from core as it happens. During replay, it reads it from the replay buffer.

Entry point: RDMSG

Classification: Subroutine

Period: None

Language: F

Activated/called by: HRPMS

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: REC — Type of record sought or −1 to return next sequential acitivity
IX — If REC = 7, type of special record sought

Common variables: Model controller messages, SUS messages, panel states, voice output indicators, etc; also RPLACT buffers, BXRAT.

Files created/changed: None

Files referenced: None

PROCEDURE RDACT

BRANCH ON CONDITION
1-PHASE 2, FREEZE AND FEED-BACK
2-REPLAY

CALL REC TO RECEIVE MESSAGE TYPE FROM ACTOUT

MESSAGE RECEIVED

READ NEXT RECORD IN ACTIVITY BUFFER

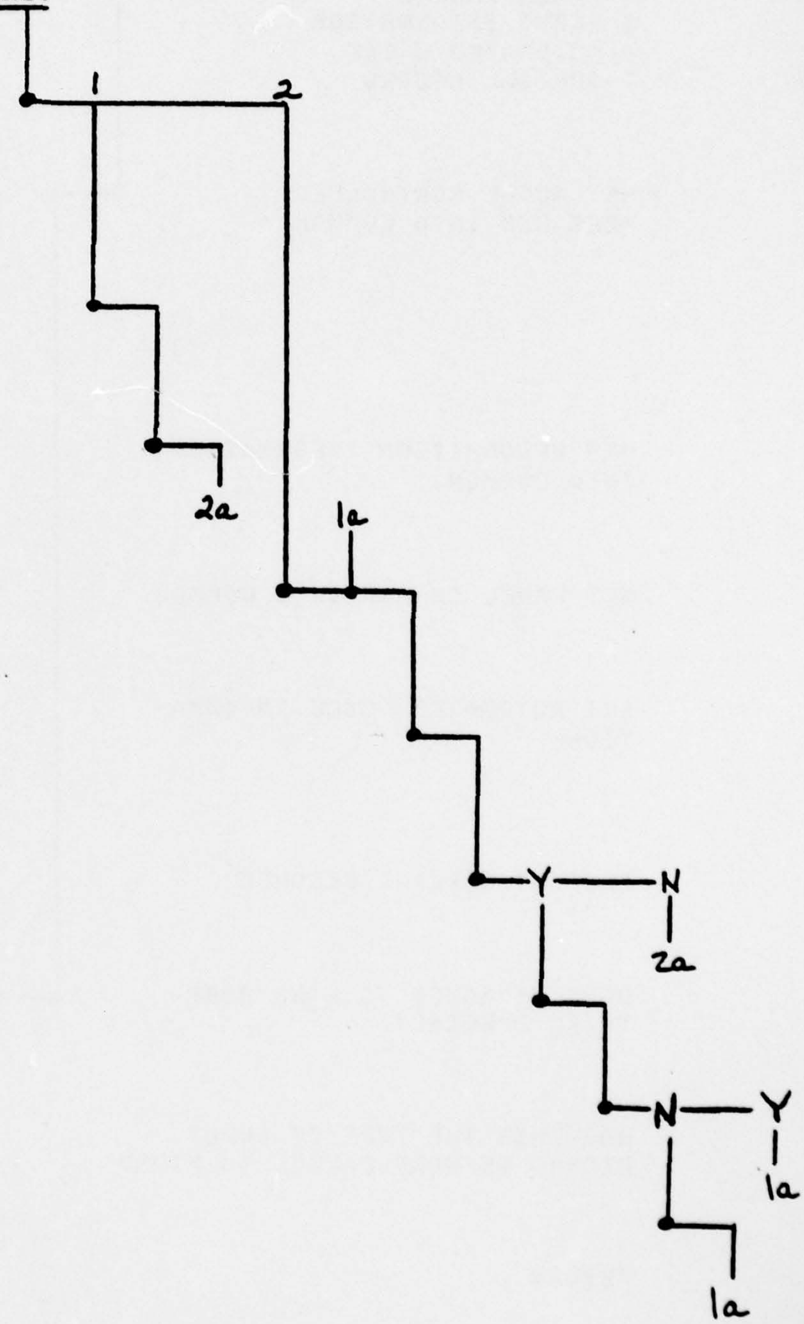
COMPUTE NEXT ADDRESS IN ACTIVITY BUFFER

END OF THIS BUFFER?

UNLOCK IT

IS NEXT BUFFER FULL?

WAIT UNTIL THERE IS MORE DATA TO READ



RDACT FLOWCHART (SHEET 1 OF 2)

BRANCH ON RECORD TYPE
2-MODEL CONTROLLER SELECTION
3-RECOGNITION BLOCK
4-PANEL CHANGE
5-SERVO INFORMATION
6-AUTOMATED VOICE
7-SPECIAL RECORD

SET MODEL CONTROLLER MESSAGES INTO COMMON

SET RECOGNITION INFORMATION INTO COMMON

SET PANEL CHANGE INTO COMMON

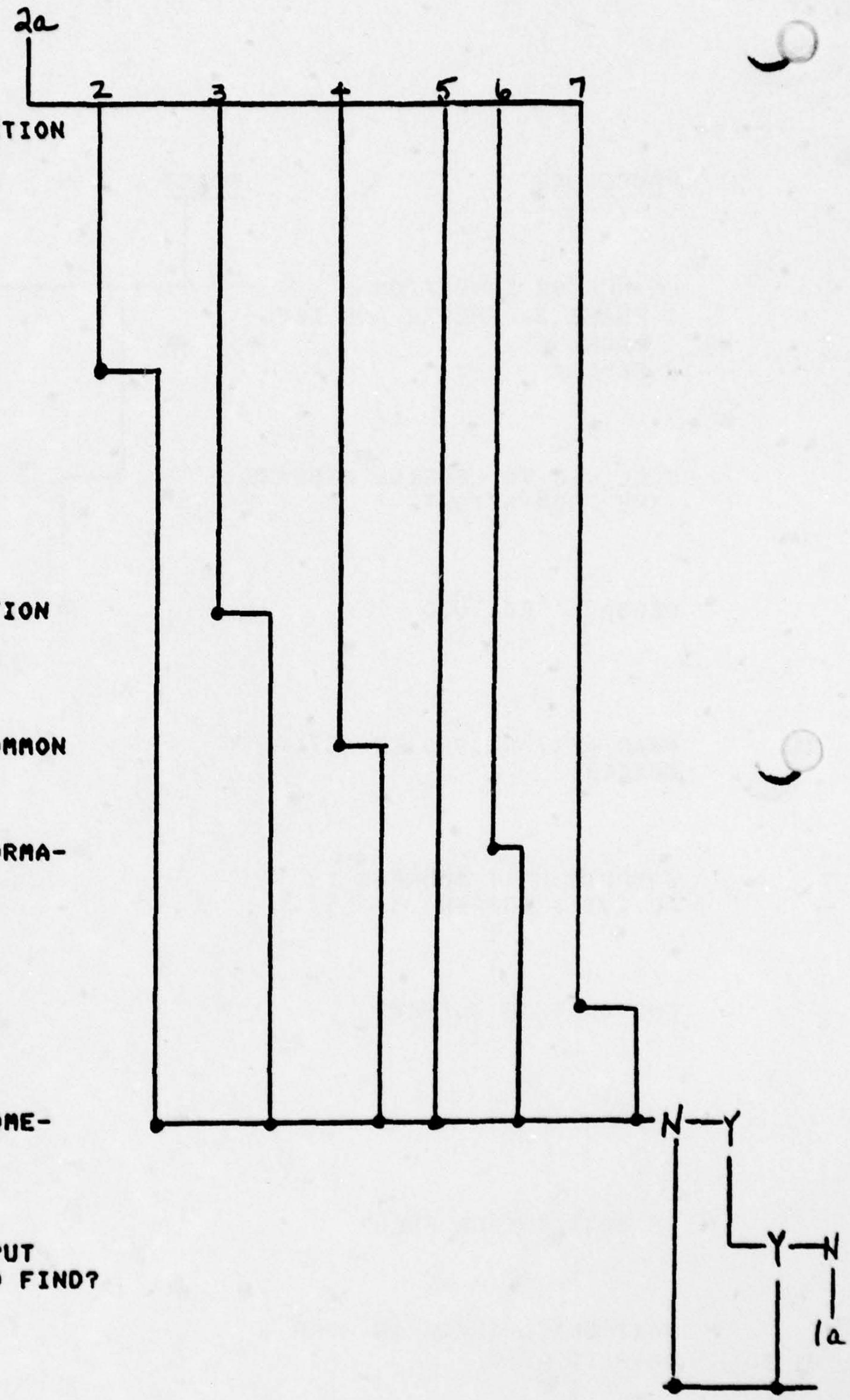
SET AUTOMATED VOICE INFORMATION

PROCESS SPECIAL RECORDS

WERE WE ASKED TO FIND SOMETHING SPECIAL?

WAS THIS THE TYPE OF INPUT RECORD WE WERE CALLED TO FIND?

RETURN



RDACT FLOWCHART (SHEET 2 OF 2)

CKADH

Description: This routine detects omission of the decision height message.

Entry point: CKADH

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKADH

Activates/calls: EXPLAIN

Cancels: None

Input arguments: MSG — The index into PFQUE

Common variables: PV09, CTRNG, PFQUE

Files created/changed: None

Files referenced: None

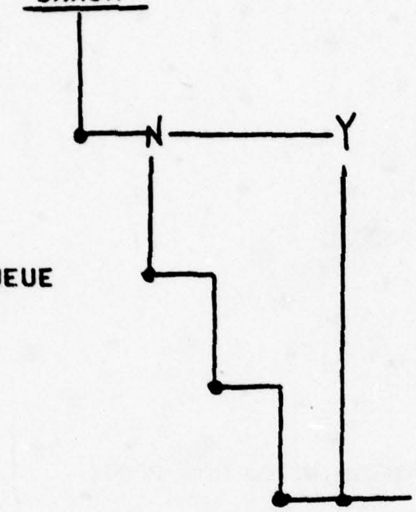
PROCEDURE CKADH

WAS ADH GIVEN?
(BIT 1 OF PV09)

REMOVE IT FROM THE QUEUE

CALL EXPLAIN

DIE



CKADH FLOWCHART (SHEET 1 OF 1)

CKAGP

Description: CKAGP checks to ensure approaching glidepath was given, and scores omissions.

Entry point: CKAGP

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKAGP

Activates/calls: EXPLAIN, CKBD through TIMSCHD

Cancels: None

Input arguments: MSG — Index into PFQUE

Common variables: PV04, CTOTHR, PFQUE

Files created/changed: None

Files referenced: None

PROCEDURE CKAGP

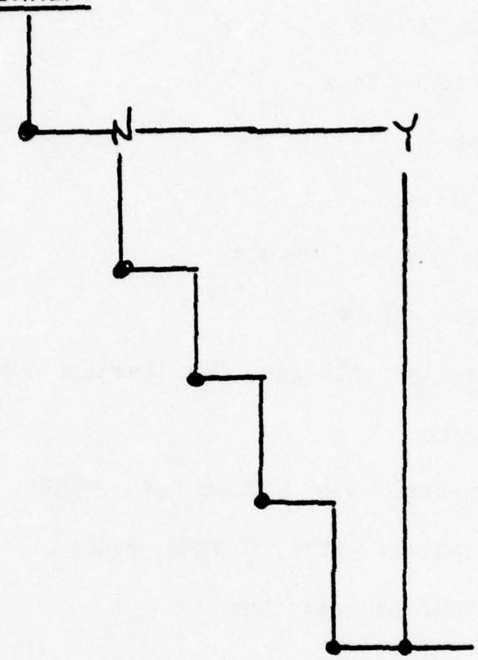
WAS ADVISORY GIVEN (BIT 0 OF PV04)?

REMOVE FROM QUEUE

CALL EXPLAIN

SCHEDULE BEGIN DESCENT CHECK

DIE



CKAGP FLOWCHART (SHEET 1 OF 1)

CKBD

Description: CKBD detects the omission of begin descent.

Entry point: CKBD

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKBD

Activates/calls: EXPLAIN

Cancels: None

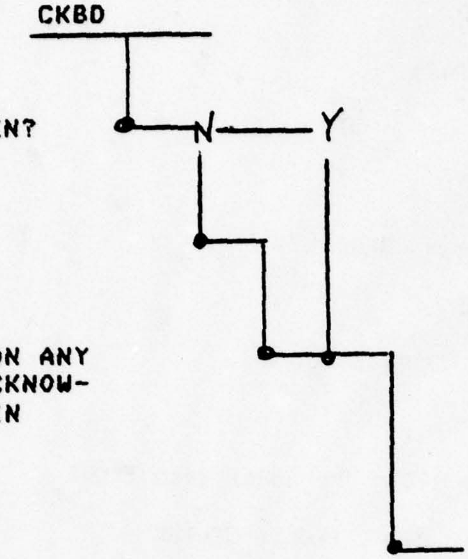
Input arguments: MSG — The index into PFQUE

Common variables: PV04, PFQUE, CTOTHR

Files created/changed: None

Files referenced: None

PROCEDURE CKBD



WAS BEGIN DESCENT GIVEN?

REMOVE IT FROM QUEUE

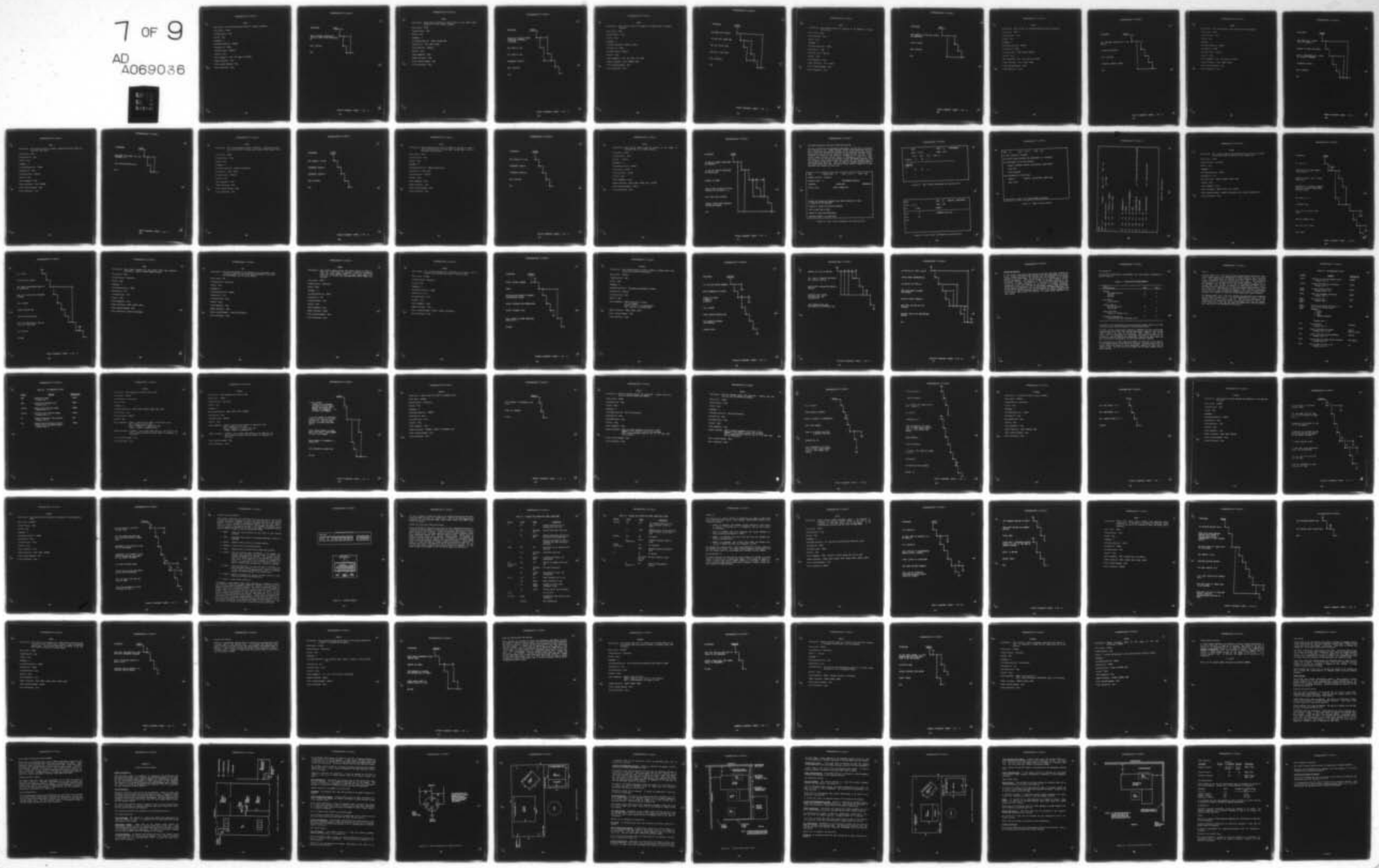
CALL EXPLAIN TO MENTION ANY
OMISSIONS OF DO NOT ACKNOW-
LEDGE, WHEELS, OR BEGIN
DESCENT

DIE

CKBD FLOWCHART (SHEET 1 OF 1)

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

7 OF 9
AD
A069036



CKCLR

Description: This routine detects failure to request clearance.

Entry point: CKCLR

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKCLR

Activates/calls: EXPLAIN

Cancels: None

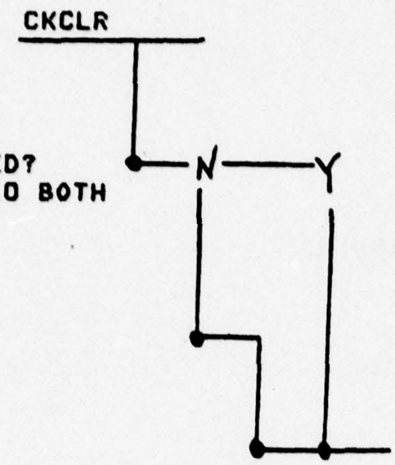
Input arguments: MSG — The index into PFQUE

Common variables: PV10

Files created/changed: None

Files referenced: None

PROCEDURE



WAS CLEARANCE REQUESTED?
(BITS 14 AND 15 OF PV10 BOTH
0)

CALL EXPLAIN

DIE

CKCLR FLOWCHART (SHEET 1 OF 1)

CKCOR

Description: CKCOR detects failures to give trend or turn within three seconds of a "well" azimuth message.

Entry point: CKCOR

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL through PV06

Cancelled by: PV05, PV06, CKCOR

Activates/calls: EXPLAIN

Cancels: None

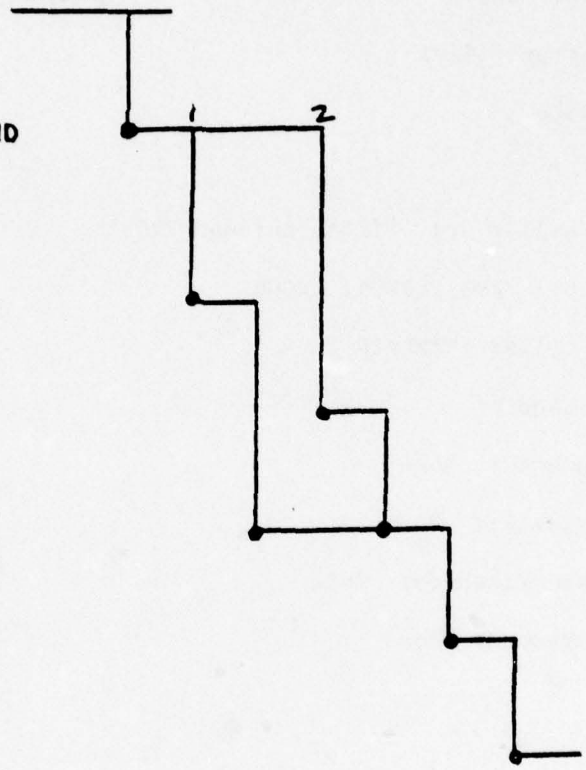
Input arguments: None

Common variables: PV06

Files created/changed: None

Files referenced: None

PROCEDURE CKCOR



BRANCH ON AZIMUTH TREND
1-HOLDING, OPENING
2-CLOSING

SET PV06(0) 1B2

SET PV06(0) 1B3

INCREMENT PV06(2)

CALL EXPLAIN

DIE

CKCOR FLOWCHART (SHEET 1 OF 1)

CKCWO

Description: This routine checks for omission of waveoff after clearance problems.

Entry point: CKWO

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL, TIMCAL

Cancelled by: CKCWO

Activates/calls: EXPLAIN

Cancels: None

Input arguments: MSG — The index into PFQUE

Common variables: PV10, CTEMERG, KYWO

Files created/changed: None

Files referenced: None

PROCEDURE CKCWO

WAS WAVE OFF GIVEN?

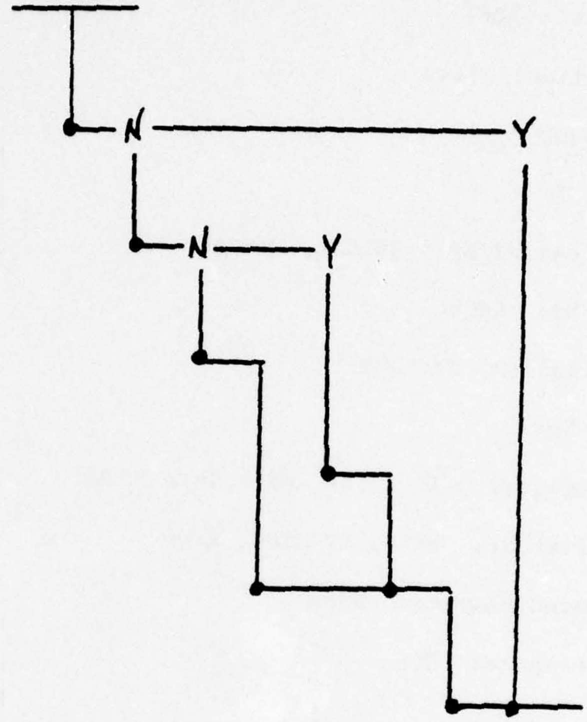
IS WAVE OFF LIGHT ON?

SET BIT 10 OF PV10

SET BIT 12 OF PV10

CALL EXPLAIN

DIE



CKCWO FLOWCHART (SHEET 1 OF 1)

CKHO

Description: This routine detects the omission of the handoff or rollout instructions.

Entry point: CKHO

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL

Cancelled by: CKHO

Activates/calls: EXPLAIN

Cancels: None

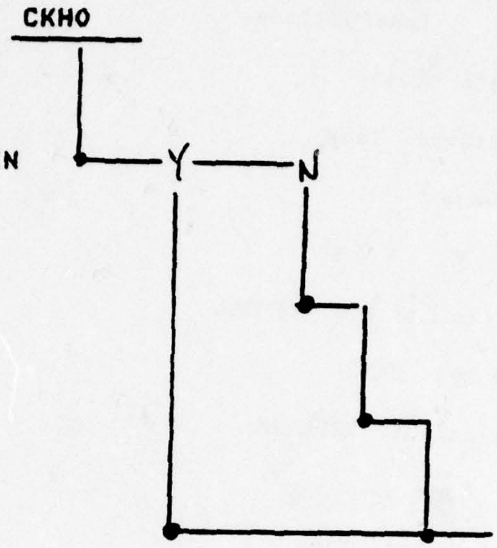
Input arguments: None

Common variables: PV12, CTOTHR

Files created/changed: None

Files referenced: None

PROCEDURE



WAS HANDOFF OR ROLLOUT GIVEN AS REQUIRED?

CLEAR CTOTHR

CALL EXPLAIN

DIE

CKHO FLOWCHART (SHEET 1 OF 1)

CKOLT

Description: CKOLT scores over landing threshold advisory omissions.

Entry point: CKOLT

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKOLT

Activates/calls: CKHO through TIMSCHD

Cancels: None

Input arguments: MSG — The index into PFQUE

Common variables: PV11, CTRNG, PFQUE

Files created/changed: None

Files referenced: None

PROCEDURE

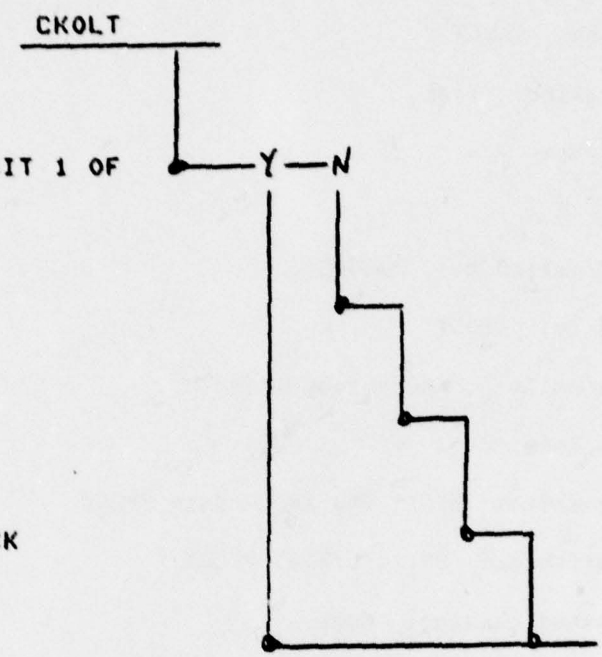
WAS ADVISORY GIVEN (BIT 1 OF PV11)?

REMOVE FROM QUEUE

CALL EXPLAIN

SCHEDULE HANDOFF CHECK

DIE



CKOLT FLOWCHART (SHEET 1 OF 1)

CKRNG

Description: This routine detects and scores mile mark omissions.

Entry point: CKRNG

Classification: Task

Period: None

Language: F

Activated/called by: RNGCAL

Cancelled by: CKRNG

Activates/calls: EXPLAIN

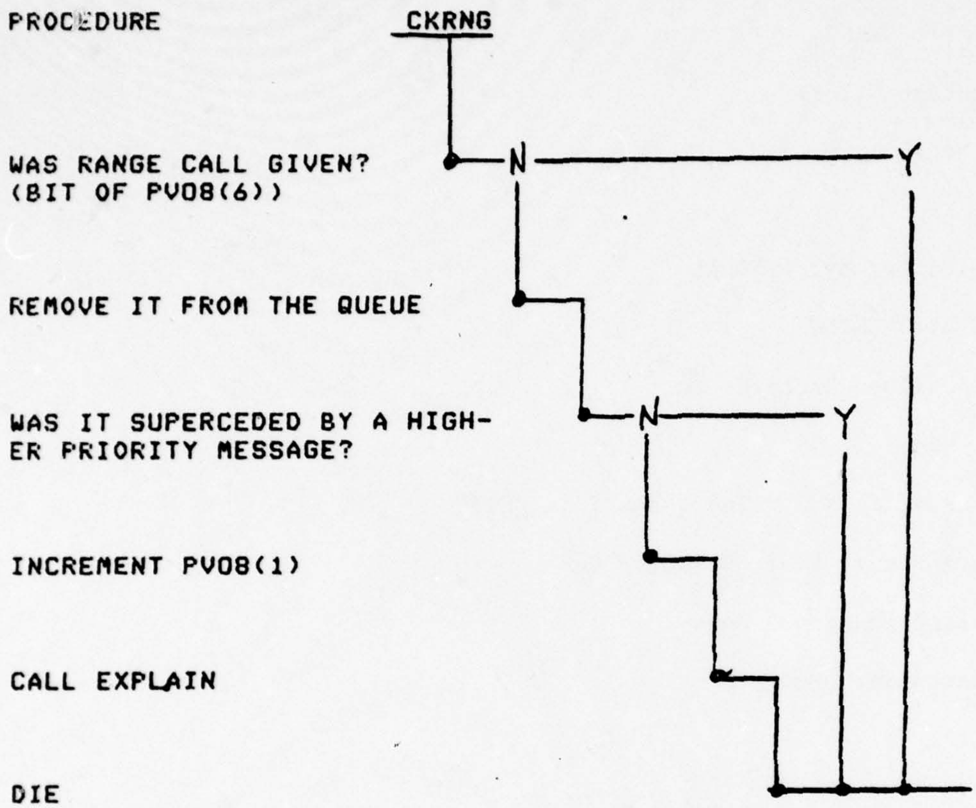
Cancels: None

Input arguments: MSG — The index into PFQUE

Common variables: PV08, CTRNG, PFQUE

Files created/changed: None

Files referenced: None



CKRNG FLOWCHART (SHEET 1 OF 1)

CKWO

Description: This routine checks to ensure a waveoff was given within the specified time limits.

Entry point: CKWO

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL

Cancelled by: CKWO

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV15, CTEMERG

Files created/changed: None

Files referenced: None

PROCEDURE CKWO

WAS WAVE OFF GIVEN (IE. IS CTEMERG -1)? N Y

CALL EXPLAIN WITH PV15

DIE

CKZN3

Description: This routine detects errors on PV05.C.3: failure to issue a correction within 20 seconds when target enters azimuth zone 3.

Entry point: CKZN3

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL through PV05

Cancelled by: PV05, CKZN3

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV05

Files created/changed: None

Files referenced: None

PROCEDURE CKZN3

SET PV05(0) TO 1B5

INCREMENT PV05(5)

INCREMENT PV05(13)

CALL EXPLAIN

DIE

CKZN3 FLOWCHART (SHEET 1 OF 1)

CK120

Description: This routine detects errors on PV05.C.2: failure to issue a counter-corrective turn within eight seconds when a turn of more than 120° is given.

Entry point: CK120

Classification: Task

Period: None

Language: F

Activated/called by: TIMCAL through PV05

Cancelled by: PV05, CK20

Activates/calls: EXPLAIN

Cancels: None

Input arguments: None

Common variables: PV05

Files created/changed: None

Files referenced: None

PROCEDURE CK120

SET PV05(0) TO 1B4

INCREMENT PV05(4)

INCREMENT PV05(12)

CALL EXPLAIN

DIE

CK120 FLOWCHART (SHEET 1 OF 1)

TGT50

Description: This routine detects when 50 percent of the target is displayed and sets a common variable.

Entry point: TGT50

Classification: Task

Period: .5 second

Language: F

Activated/called by: PM*INT

Cancelled by: TGT50

Activates/calls: ACTOUT

Cancels: None

Input arguments: None

Common variables: GZPHZ, BXPV, ACRNG, PV01, CLTGT50

Files created/changed: RPLACT

Files referenced: None

PROCEDURE

IS 50% OF TARGET DISPLAYED
ON AZIMUTH?

IS 50% OF TARGET DISPLAYED
ON ELEVATION?

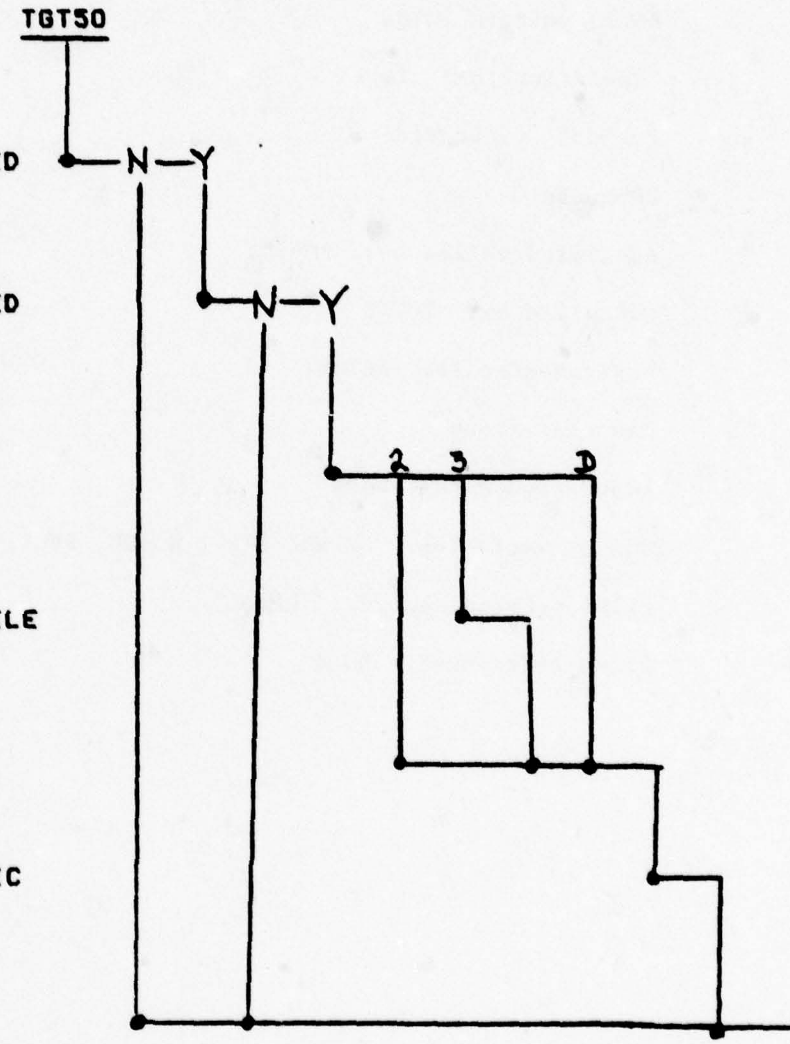
BRANCH ON PHASE

WRITE TIME IN ACTIVITY FILE
(ACTOUT(7,1,CLTICK,....))

SET TIME INTO CLTGT50

REMOVE TGT50 FROM PERIODIC
SCHEDULING QUEUE

DIE



TGT50 FLOWCHART (SHEET 1 OF 1)

Performance Evaluation and Data Formatting Routines

The data collected by the performance measurement routines must be evaluated and formatted for use. These routines compute scores based upon the student's performance and provide feedback to the instructor and the student. These scores also provide the basis for the various types of status reports available on request by the instructor (figures 36, 37, and 38). These status reports provide various levels of information, down to an exact explanation of the trainee's performance on every performance variable. The more complete status reports show the reason for the training system's problem selection decisions. Figure 39 shows the feedback normally given to the student after every run. The replay report shown in figure 40 is also available upon request for every run.

NAME: PRESENT TASK: 6.4 DATE: 4-10-79 TIME: 1058

CURRENT ACTIVITY: PROBLEM 5

PREVIOUS TASK: 3.2 PERFORMANCE ANALYSIS:

STRENGTHS	BORDERLINE	WEAKNESSES
TURN TO FINAL	COURSE CORRECTIONS	.
.	.	.
.	.	.
.	.	.

STUDENT WAS ADVANCED TO PRESENT LEVEL AFTER COMPLETING 15 RUNS
NO REMEDIATION WAS NECESSARY

DIAGNOSIS: HEADING STYLIZATION PROBLEMS

TOTAL SYSTEM TIME TO DATE:

PERCENT OF TIME USED EFFECTIVELY:

PREDICTED COURSE COMPLETION TIME:

Figure 36. Type 1 Status Information for Instructor Use

NAME: TASK: 3.2 PERFORMANCE

DATE: 4-10-79 TIME: 1058

	PV01	PV02	PV03	PV04......
NOMINAL	70	80	.....	
OBSERVED RUN				
1	90	85	.....	
2	.	.		
.	.	.		
.	.	.		

Figure 37. Type 2 Status Information for Instructor Use

NAME: TASK: 3.2 PROBLEM 7 PERFORMANCE

DATE: 4-10-79 TIME: 1058

PMV	SCORE	ERRORS
PV01	95	INCORRECT CALL SIGN
PV02	100	
PV03		
.		
.		
.		

Figure 38. Type 3 Status Information for Instructor Use

```
TASK:  3.2          DATE:  5-10-79      TIME:  1058

YOU HAVE COMPLETED 5 PROBLEMS

THE MINIMUM NUMBER REQUIRED FOR ADVANCEMENT IS 10 PROBLEMS

YOUR PERFORMANCE ON THE NEW MATERIAL

     REQUEST CLEARANCE (PERFECT, SATISFACTORY, NEEDS WORK)

     ISSUE WIND             .

     ISSUE CLEARANCE        .

YOUR PERFORMANCE ON OTHER TASKS

     HANDOFF:          (PERFECT, SATISFACTORY, NEEDS WORK)

     RADIO CHECK            .

         .                  .

         .                  .

         .                  .

         .                  .

SUGGESTIONS:  PAUSE A BIT LONGER BETWEEN ADVISORIES
```

Figure 39. Sample Student Feedback

NAME:

TASK: 4.5 PROBLEM: 4 DATE: 5-10-79 TIME: 1058

REC	SPKR	RANGE	TIME	ADVISORY	REF	ERROR
1	PTN	8.0	0	POSITION 4....HANDOFF		
2	GCA	7.9	7	POSITION 4, ROGER		
.						
.						
30	GCA	5.7	50	APPROACHING GP		
33	PLT	5.6	53	ROGER		
34	GCA	5.4	60	C/S, DO NOT ACKNOWLEDGE:...		
37	GCA	5.4	62	OVER		
41	GCA	5.1	70	SLIGHTLY BELOW GP	X.X	GP MSGS NOT LEGAL PRIOR TO BEGIN DESCENT
44	GCA	5.1	73	BEGIN DESCENT		
47	GCA	4.9	80	?		
50		4.9	82	ADVISORY OMISSION	X.X	5 MILES MUST BE GIVEN
.						
.						
.						

Figure 40. Replay Report

SCORE

Description: This routine computes the quantitative score for each scored PMV and stores the information in the student file.

Entry point: SCORE

Classification: Subroutine

Period: None

Language: F

Activated/called by: P3TRM

Cancelled by: N/A

Activates/calls: SELECT, MISMOD, PRRPT, FBAK

Cancels: None

Input arguments: None

Common variables: PVN**, PVE**, PV**, PFS**

Files created/changed: Student performance file, student feedback file

Files referenced: None

PROCEDURE SCORE

SET PMV TO 1

WAS THIS PMV TO BE SCORED
(PVN**.NE.-1)?

COMPUTE SCORE, PVE**, BASED
UPON PV**

CONSTRUCT A FEEDBACK MESSAGE
BASED UPON ERROR SCORE AND
NOMINAL VALUE

SET PVE** TO -1

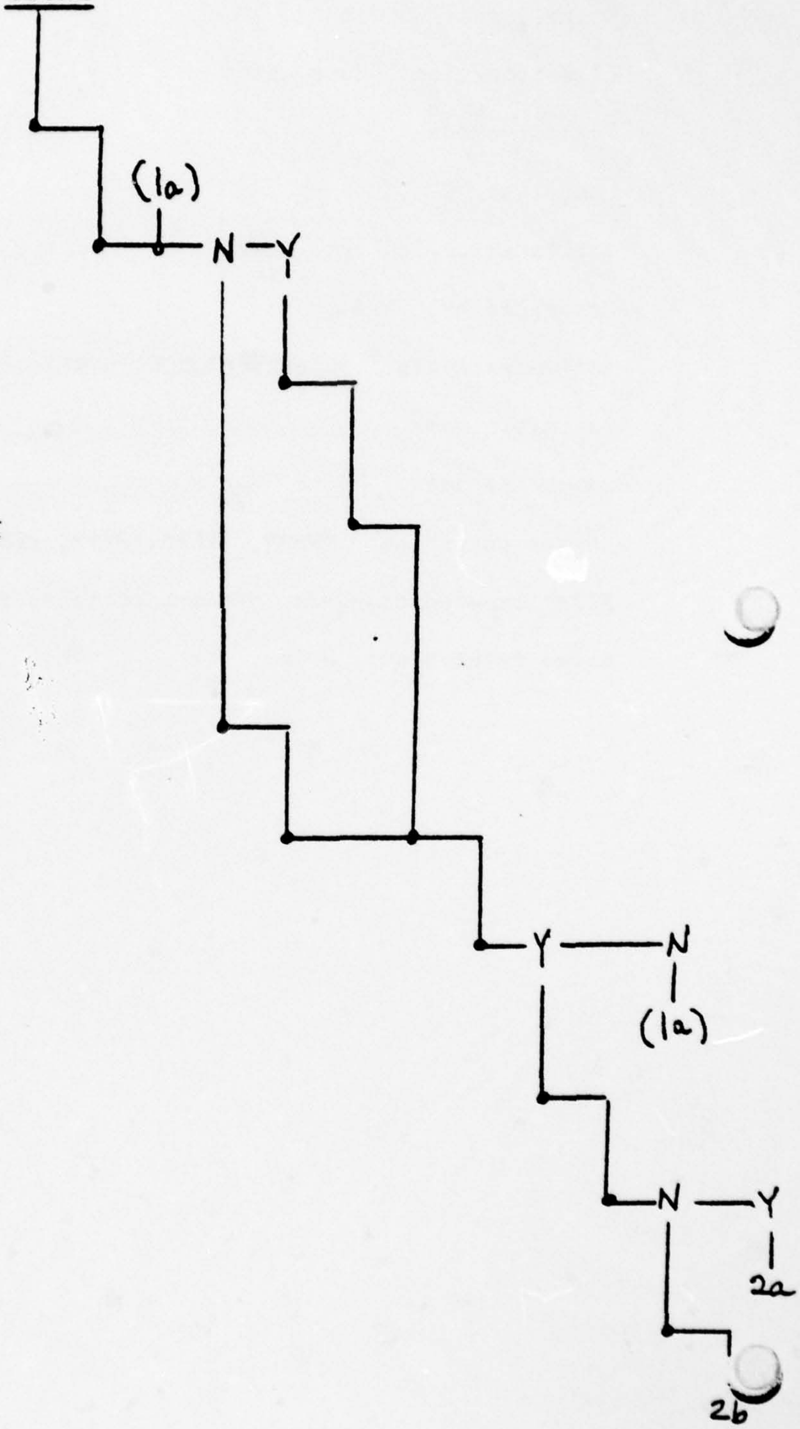
INCREMENT PMV

WAS THIS THE END OF THE
PMVS?

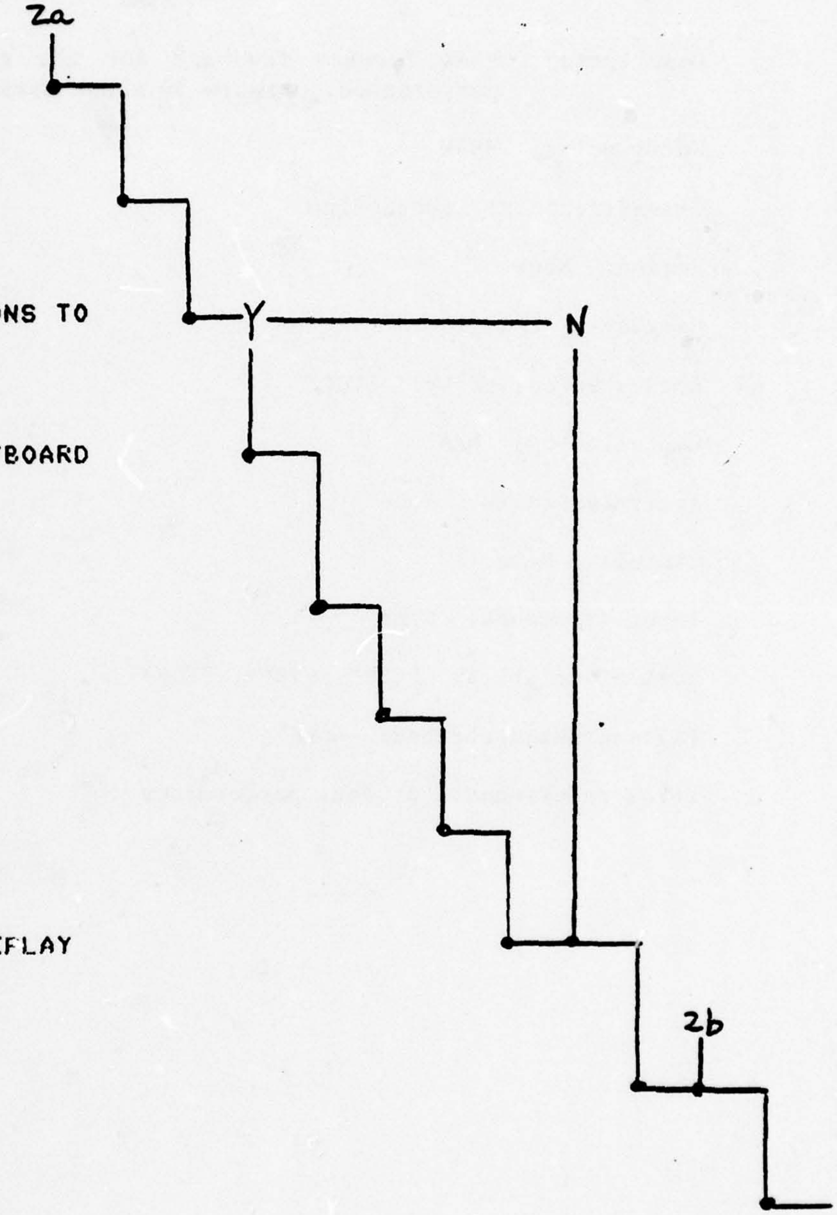
UPDATE STUDENT FILE

WAS THIS THE P-RUN?

CALL FBAK



SCORE FLOWCHART (SHEET 1 OF 2)



CALL PRRPT

PRINT REPLAY REPORT

ARE THERE MISRECOGNITIONS TO
BE RESOLVED?

WAIT FOR INSTRUCTOR KEYBOARD
REQUEST

CALL MISMOD

SCORE REVISED RUN

PRINT REVISED REPORT

WAIT FOR DELETION OF REPLAY
FILE BY INSTRUCTOR

CALL SELECT

RETURN

SCORE FLOWCHART (SHEET 2 OF 2)

FBAK

Description: FBAK formats feedback for the student based upon observed performance. Figure 39 shows sample outputs.

Entry point: FBAK

Classification: Subroutine

Period: None

Language: F

Activated/called by: SCORE

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: PVN**, PVE**, PFS**

Files created/changed: None

Files referenced: Student performance

TFB

Description: TFB provides feedback to the instructor on a task basis; (type 1 report, figure 36) also notes this information in the student performance file for use by STATUS.

Entry point: TFB

Classification: Subroutine

Period: None

Language: F

Activated/called by: SELECT

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: None

Files created/changed: Student performance

Files referenced: None

PRRPT

Description: This routine simply prints the replay report as shown in figure 40. This report is optional. After the performance test, the record numbers shown on this report enable the instructor to correct any misrecognitions which might have occurred.

Entry point: PRRPT

Classification: Subroutine

Period: None

Language: F

Activated/called by: SCORE

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: NCLPT

Files created/changed: None

Files referenced: None

MISMOD

Description: This routine modifies the performance test replay file to resolve any misrecognitions prior to final scoring.

Entry point: MISMOD

Classification: Subroutine

Period: None

Language: F

Activated/called by: SCORE

Cancelled by: N/A

Activates/calls: None

Cancels: None

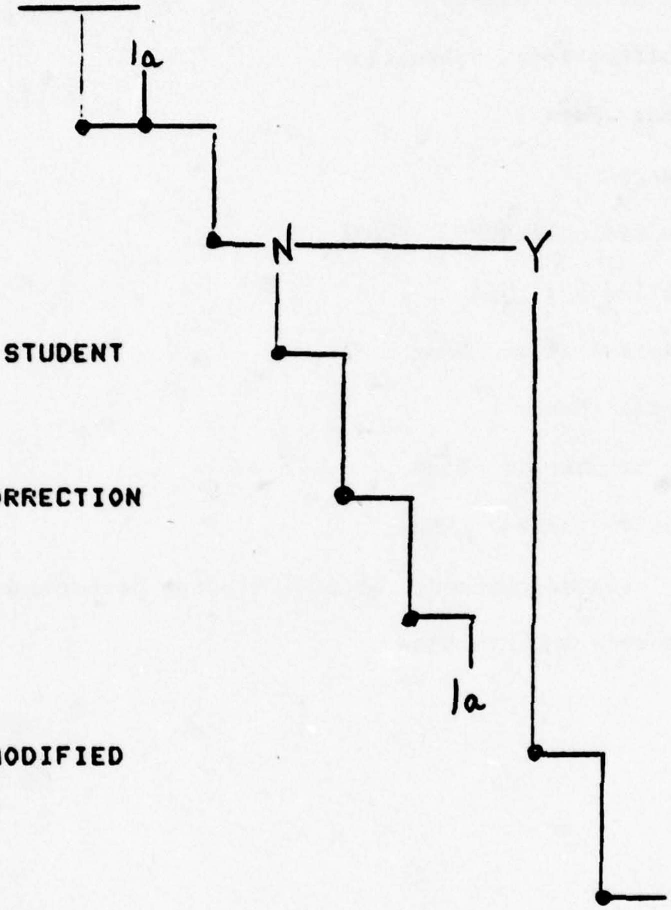
Input arguments: None

Common variables: None

Files created/changed: RPLACT, student performance

Files referenced: None

PROCEDURE MISMOD



ACCEPT RECORD NUMBER

DONE?

RETRIEVE AND DISPLAY STUDENT PERFORMANCE RECORD

ACCEPT RECOGNITION CORRECTION

MODIFY STUDENT FILE

CALL SCORE TO SCORE MODIFIED REPLAY FILE

RETURN

MISMOD FLOWCHART (SHEET 1 OF 1)

EXPLAIN

Description: This routine freezes on error in phase 2, freezes type 4 and 5 replays and provides explanations.

Entry point: EXPLAIN

Classification: Task

Period: None

Language: F

Activated/called by: Performance measurement routines

Cancelled by: EXPLAIN

Activates/calls: None

Cancels: None

Input arguments: TIME — Time error occurred
PV — PMV number
START — first bit of offending PMV
END — last bit of offending PMV

Common variables: GZPHZ, GZRPL, GZFRZ

Files created/changed: None

Files referenced: None

PROCEDURE

IS THIS PMV BEING SCORED?

SAVE ARGUMENTS ON STACK

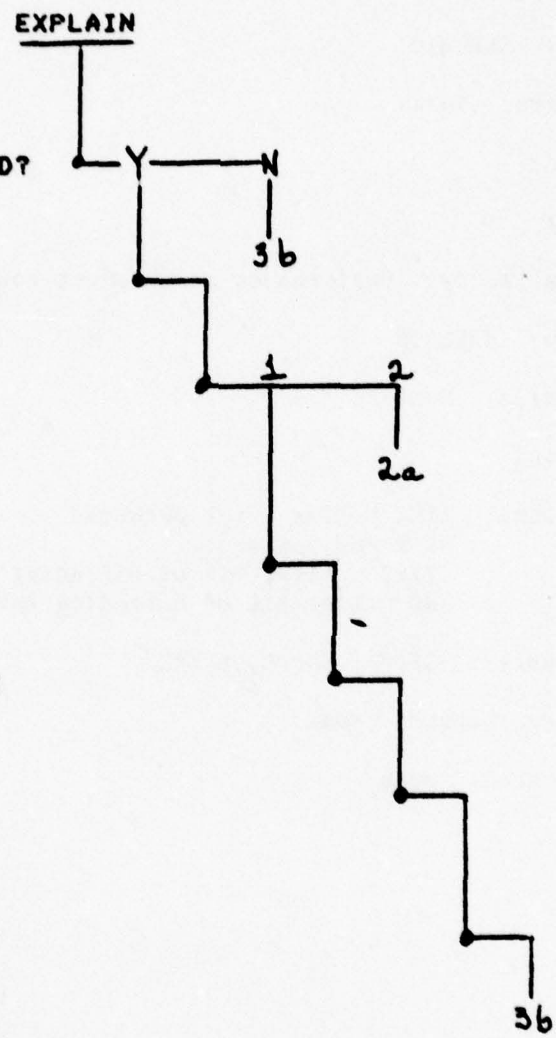
BRANCH ON PHASE
1-PHASE 2
2-REPLAY

CALL FREEZE

SPEAK ERROR DESCRIPTION

SET FREEZE ON ERROR
TERMINATION

WAKEUP EVPHZ



EXPLAIN FLOWCHART (SHEET 1 OF 3)

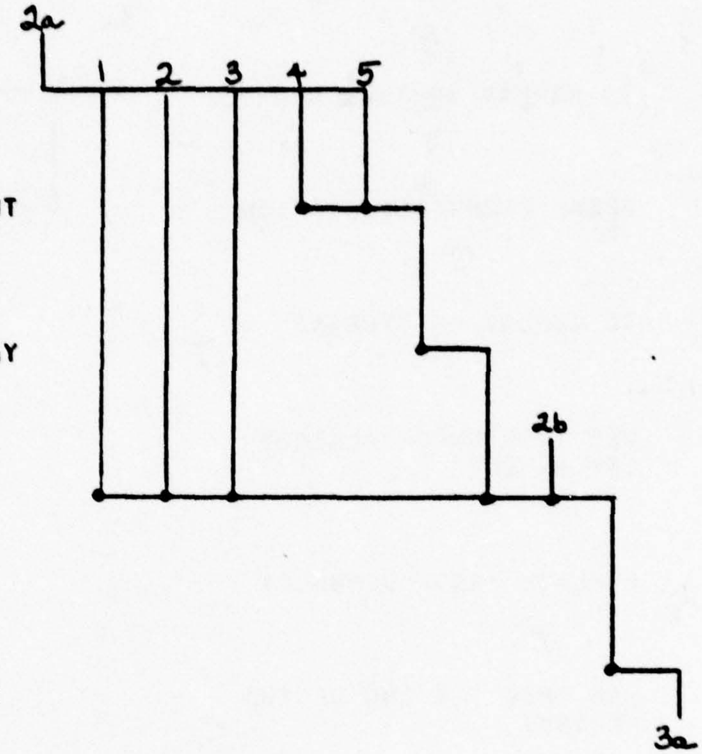
BRANCH ON TYPE OF REPLAY

SET TIME TO FREEZE FOR SPOUT
ETC (RPTSP, RPTDLY)

WAIT EVSPT INDICATING REPLAY
FROZEN

RETRIEVE NEXT ERROR
DESCRIPTION, AND
EXPLANATION

NOTE DESCRIPTION AND
EXPLANATION IN SCORING FILE



EXPLAIN FLOWCHART (SHEET 2 OF 3)

IS REPLAY OF TYPE 4 OR 5?

SPEAK ERROR DESCRIPTION

IS REPLAY OF TYPE 5?

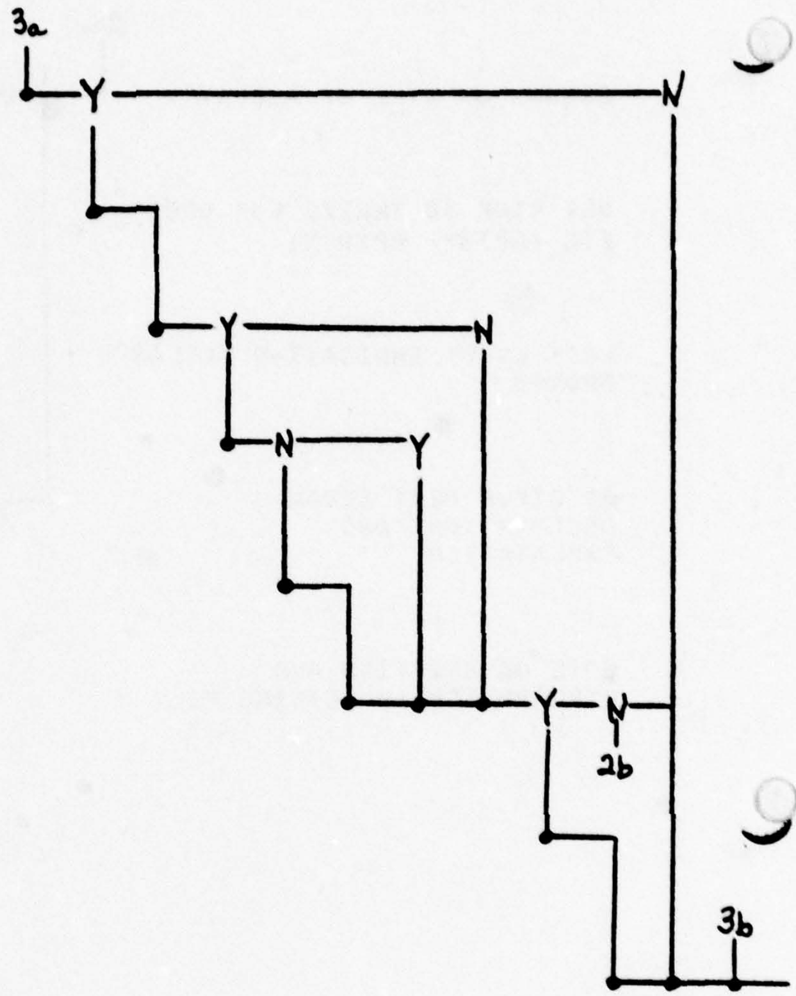
WAS THIS ERROR ALREADY
EXPLAINED?

EXPLAIN ERROR VERBALLY

WAS THIS THE END OF THE
ERRORS?

RESTART REPLAY BY RESTARTING
RPCLOK

DIE



EXPLAIN FLOWCHART (SHEET 3 OF 3)

Supporting Routines

In this section functional descriptions of various supporting routines are given. Program descriptions are included where they contribute to an understanding of the processing. The majority are I/O interfaces to the various GCA-CTS peripherals. They can be thought of as constituting a GCA-CTS environment library. The existence of the I/O interface routines in particular is of utmost importance in a multitask environment where many asynchronous tasks must compete for the control of a single peripheral. Descriptions are provided for IPB I/O processing, keyboard input processing, Votrax (speech synthesizer), I/O, record/playback (or speech digitizer) I/O, as well as for special functions such as activity file output, range-related task queuing, time-related task queuing, the random number generator, and user clocks.

Initialization

The GCA-CTS initialization requirements fall into several categories as shown in table 37.

TABLE 37. INITIALIZATION REQUIREMENTS

Type of Initialization	CPU1	CPU2
Startup		
IPB communications	X	X
Hardware	X	
Software	X	X
New Student		
File creation		X
Student sign on		
Problem selection	X	
File access	X	X
Phase initiation (Phase 1-3, Replay, etc.)	X	X
Problem initialization (APE variables, PMS variables, etc.)	X	X

In general, the initialization routines prepare hardware devices and disk files for access, and set variables to their initial values.

For CPU1, startup initialization (handled by SYSINIT) includes establishing communications with CPU2, and setting appropriate lights on the trainee panel to reflect the state of the buttons. When a student signs on the system, CPU1 is responsible for retrieving the student's records and for selecting a training problem for presentation. Each of the phase executives has initialization requirements, as does each individual problem.

The initialization of CPU2 (handled by TUNIT) is restricted to the creation of the Megatek display list and servo controls. Pictures will be set to initial values and turned off until appropriate requests are made from the IMAGES routine. The servo will be activated, but not under student control until required.

IBP I/O

To make optimal use of the multiprocessor design while maintaining acceptable response time, good communication between systems 1 and 2 is essential. This communication is handled by various routines which send and receive information from the interprocessor bus (IPB). The information passed consists of a message code and up to eight arguments. Some or all of the arguments will be sent to the task or subroutine designated by the message code. CPU2 receives information on target position, servo control and speech-related messages such as start/stop speech understanding or voice data collection. CPU1 receives information on servo position, recognition blocks from the speech recognition program to be sent to speech understanding, VDC requests for audio and Votrax prompts and keyboard wakeups.

IPB messages begin with a code which defines the destination of the message and the number of words in it. All data transferred are binary. The IPB messages are shown in table 38 and 39. The routines which handle this communication are IPBOUT1 and IPBOUT2, subroutines in CPU1 and CPU2 respectively which write to the IPB, and IPBIN1 and IPBIN2 which read the messages and convey them to the appropriate destinations.

TABLE 38. IPB MESSAGES TO CPU2

SOURCE	CONTENT	DESTINATION
PHAZ*	Requests for picture display control Message code = 2	IMAGES
SYSINIT	System initialization information Message code = 1	TUNIT
RADAR PHAZ*	Aircraft display updates Message codes = 3,4	PICUP
RADAR	Servo and alignment information Message code = 5	SERVO
PHAZ* REPLAY PRUN	File name of text Message code = 6	TEXT
PHAZ* REPLAY PRUN	Retrieve and display next page, or one of the following canned messages: 1 erase 2 break 3 home . feedback messages . . Message code = 7	TEXT
TZEC	Menu template Message code = 8	SYS2CTRL
PHAZ*	Speech validation and update Message codes = 9,10	Speech Wake up events
ISAY	Model controller selected messages Message code = 11	Hearsay
PHAZ*	Type string on trainee station teletype Message code = 12	CRT handler
PHAZ*	Kill yourself, return to CLI Message code = 13	All

TABLE 39. IPB MESSAGES TO CPU1

SOURCE	CONTENT	DESTINATION
KBRD	Subdirectory name Message code = 1	KPROC
KBRD	Process this keyboard entry Message code = 1	KPROC
Display	Azimuth servo position change Message code = 5	RADAR
Display	Elevation servo position change Message code = 5	RADAR
SR	Message recognized, time completed Message code = 2	SUS
VDC	Prompts from voice data collection, requests audio and VOTRAX prompts Message code = 3	VSCON

IPBOUT1

Description: Sends messages across the IPB to CPU2.

Entry point: IPBOUT1

Classification: Subroutine

Period: None

Language: F

Activated/called by: RADAR, PHAZ*, REPLAY, PRUN, TZEC, ISAY

Cancelled by: None

Activates/calls: ARGCHK

Cancels: None

Input arguments: TASKID — Identification number of destination task
MSG — message code for task
STUFF — arguments for destination task
IPARGNO — number of arguments sent

Common variables: IPOUTNO1 — array whose index refers to the task ID, and whose contents signify the valid number of arguments for that task

Files created/changed: None

Files referenced: None

IPBOUT2

Description: Sends messages across IPB to CPU1.

Entry point: IPBOUT2

Classification: Subroutine

Period: None

Language: F

Activated/called by: KBRD, SERVO, SAID, PRESENT

Cancelled by: None

Activates/calls: ARGCHK

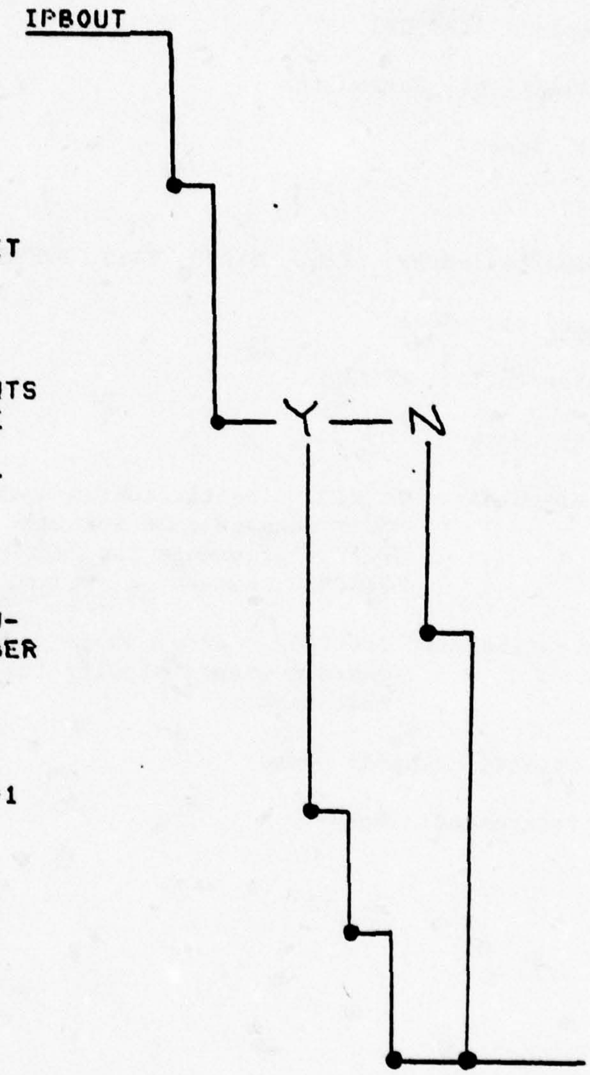
Cancels: None

Input arguments: TASKID — identification number of destination task
MSG — message code for task
STUFF — arguments for destination task
IPARGNO — number of arguments set

Common variables: IPOUTNO2 — array whose index refers to the task ID, and whose contents signify the valid number of arguments for that task

Files created/changed: None

Files referenced: None



X CALL ARGCHK
(ARGHK IS AN ASSEMBLY LANGUAGE ROUTINE WHICH CHECKS THE STACK FOR NUMBER OF ARGUMENTS. IT RETURNS THE NUMBER)

X IS THE NUMBER OF ARGUMENTS CORRECT FOR THE TASK TO BE ACTIVATED? (DOES # FROM ARGCHK= IPOUTNO1(TASK NUMBER)?)

TYPE-"WRONG NUMBER OF ARGUMENTS FOR TASK " TASK NUMBER "ON CPU" CPU NUMBER

SEND NUMBER OF ARGUMENTS +1 ACROSS IPB

SEND ARGUMENTS ACROSS IPB

RETURN

IPBOUT FLOWCHART (SHEET 1 OF 1)

ARGCHK

Description: Checks stack for number of arguments sent.

Entry point: ARGCHK

Classification: Subroutine

Period: None

Language: A

Activated/called by: IPBOUT*

Cancelled by: None

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: IPARGNO — Number of arguments sent

Files created/changed: None

Files referenced: None

ARGCHK

TAKE NUMBER OF ARGUMENTS OFF STACK

STORE IN IPARGNO

RETURN

ARGCHK FLOWCHART (SHEET 1 OF 1)

IPBIN1

Description: Receives messages across IPB from CPU2. Inserts them into array for use by TALKOUT and LOOKOUT.

Entry point: IPBIN1

Classification: Task

Period: None

Language: F

Activated/called by: CPU1 initialization

Cancelled by: None

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: ARGS — contains arguments to be sent to tasks
TASKORD — lists tasks in order to be processed
FREE — index indicates task ID, set to -1 if task ready for activation

Files created/changed: None

Files referenced: None

IPBIN2

Description: Receives messages across IPB from CPU1. Inserts them into array for use by TALKOUT and LOOKOUT.

Entry point: IPBIN2

Classification: Task

Period: None

Language: F

Activated/called by: CPU2 initialization

Cancelled by: None

Activates/calls: None

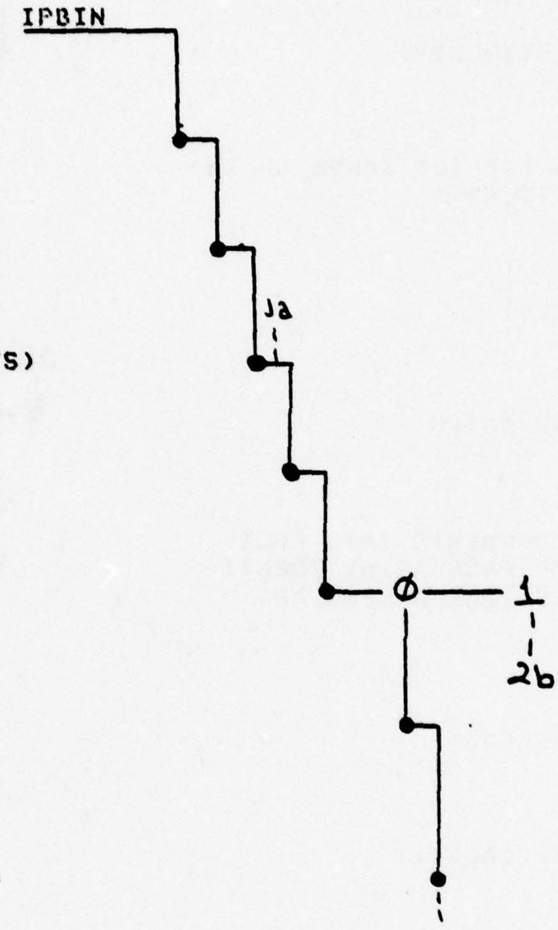
Cancels: None

Input arguments: None

Common variables: ARGS — contains arguments to be sent to task
TASKORD — lists tasks in order to be processed
FREE — index indicates task ID, set to -1 if task ready for new information

Files created/changed: None

Files referenced: None



CALL SETARAY

TASK TALKOUT,LOOKOUT

READ N (NUMBER OF ARGUMENTS)

READ TASK NUMBER

TASK IS 0-SPEECH RELATED
1-DISPLAY RELATED

SUSPEND 20,40

READ ARGUMENTS INTO POSITION IN ARGS ARRAY IDENTIFIED BY TASK NUMBER AND BQTALK

IPBIN FLOWCHART (SHEET 1 OF 2)

BQTALK=BQTALK+1

TIDLST=TIDLST+1

IF TIDLIST TOO LARGE AN IN-
DEX, TIDLST=1

ID=TIDLST

SUSPEND 30,40

READ ARGUMENTS INTO POSI-
TION IN ARGS ARRAY IDENTI-
FIED BY TASK NUMBER AND
SEEBQ

SEEBQ=SEEBQ+1

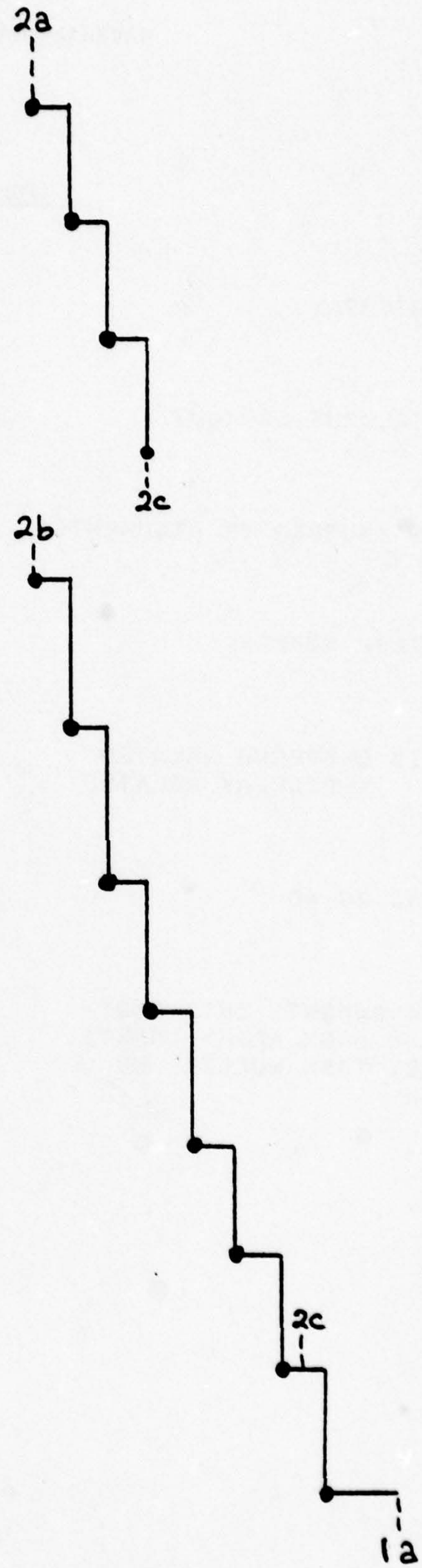
PIDLST=PIDLST+1

IF PIDLST TOO LARGE AN INDEX
PIDLST=1

ID=PIDLIST

TASKORD(ID)=TASK NUMBER

WAKEUP 40



IPBIN FLOWCHART (SHEET 2 OF 2)

SETARAY

Description: Initializes arrays for use by IPBIN*.

Entry point: SETARAY

Classification: Subroutine

Period: None

Language: F

Activated/called by: IPBIN*

Cancelled by: None

Activates/calls: None

Cancels: None

Input arguments: None

Common variables: ARGS, TASKORD, FREE

Files created/changed: None

Files referenced: None

SETARY

SET FREE ARRAY TO -1

SET ARGS ARRAY TO -1

SET TASKORD ARRAY TO -1

RETURN

SETARY FLOWCHART (SHEET 1 OF 1)

LOOKOUT

Description: Sends display-related messages and arguments to the appropriate task.

Entry point: TALKOUT

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN*

Cancelled by: None

Activates/calls: None

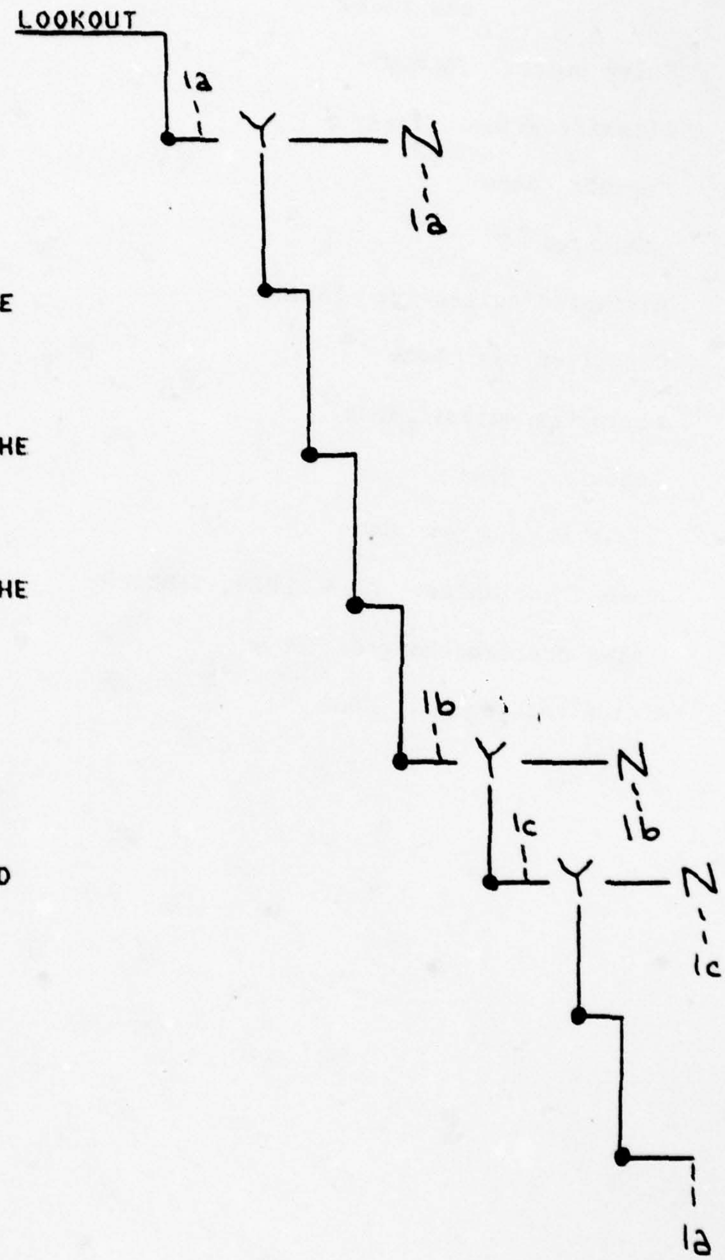
Cancels: None

Input arguments: Nqne

Common variables: ARGS, FREE, TASKORD

Files created/changed: None

Files referenced: None



ARE WE READY TO ACTIVATE ANOTHER TASK?

SET THE DONE FLAG FOR THE LAST TASK ACTIVATED IN FREE ARRAY

INCREMENT THE POINTER TO THE TOP OF THE QUEUE

INCREMENT THE POINTER TO THE ARRAY CONTAINING THE ORDER OF THE TASKS (LOOKORD)

IS THERE ANOTHER TASK?

IS THE TASK TO BE ACTIVATED READY FOR ACTIVATION?

SET THE BUSY FLAG FOR THE TASK IN FREE

SEND THE ARGUMENTS TO THE APPROPRIATE TASK

LOOKOUT FLOWCHART (SHEET 1 OF 1)

TALKOUT

Description: Sends speech-related messages and arguments to the appropriate tasks.

Entry point: TALKOUT

Classification: Task

Period: None

Language: F

Activated/called by: IPBIN*

Cancelled by: None

Activates/calls: None

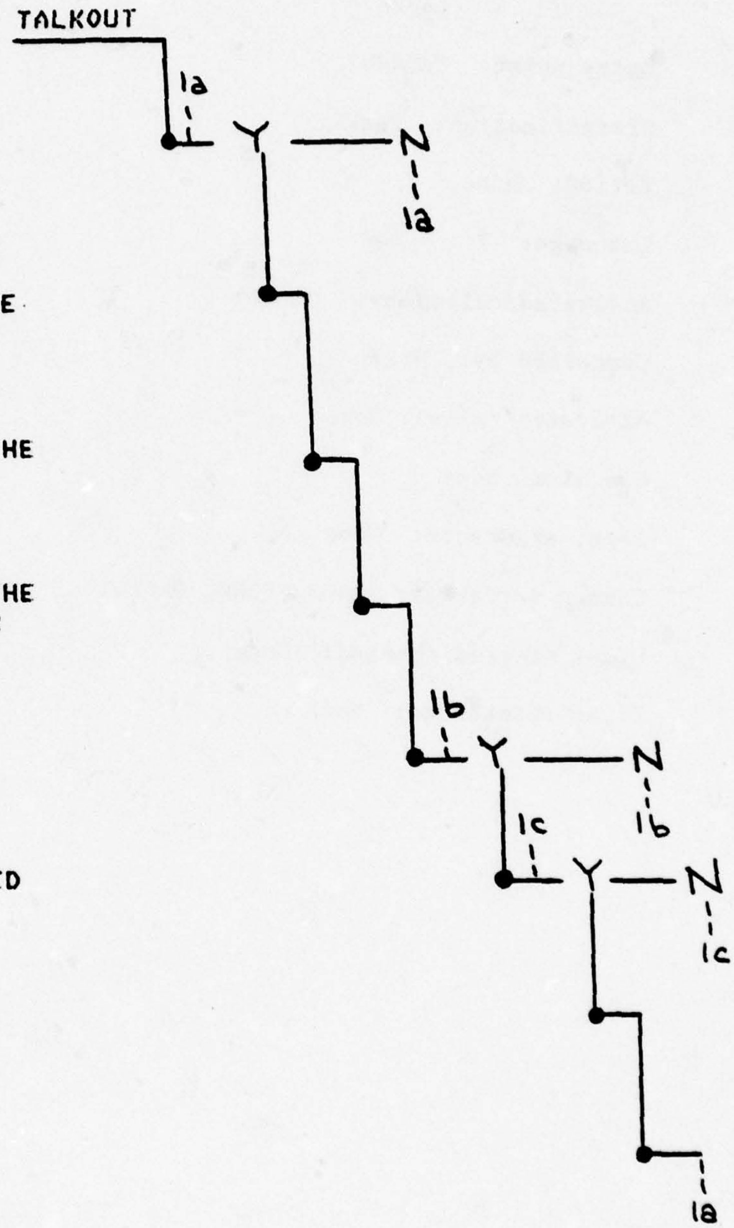
Cancels: None

Input arguments: None

Common variables: ARGS, FREE, TASKORD

Files created/changed: None

Files referenced: None



ARE WE READY TO ACTIVATE ANOTHER TASK?

SET THE DONE FLAG FOR THE LAST TASK ACTIVATED IN FREE ARRAY

INCREMENT THE POINTER TO THE TOP OF THE QUEUE

INCREMENT THE POINTER TO THE ARRAY CONTAINING THE ORDER OF THE TASKS (TALKORD)

IS THERE ANOTHER TASK?

IS THE TASK TO BE ACTIVATED READY FOR ACTIVATION?

SET THE BUSY FLAG FOR THE TASK IN FREE

SEND THE ARGUMENTS TO THE APPROPRIATE TASK

TALKOUT FLOWCHART (SHEET 1 OF 1)

Keyboard Input Processing

Both student controller and instructor will have the use of the GCA-CTS keyboard to furnish information or modify the training process. The GCA-CTS keyboard features a standard typewriter keyboard with an 11-key numeric keypad. It also includes 11 user-defined special function keys, nine of which are used in the training process. All keyboard processing except validity checking as handled by the CPU1 routine KBRD. The special function keys are shown in figure 41 and are defined as follows:

1. MENU — Print out valid entries for this point in the training process.

2. CHAL — Challenge this section of training material, proceed to grading.

3. REPEAT — Repeat previous portion of training session.

4. PROCEED — Continue with the training session.

5. ALIGN — Align azimuth, elevation and/or range mark displays.

6. STATUS — This key provides basic information to the student controller while allowing the instructor to request more detailed information as desired. The student receives data on his present position in the syllabus, his present task, and, if in phase 3, the number of problems completed. In addition to this, the instructor receives a menu of further status information to access if required.

7. SIGNON — Cause initialization to occur for new or returning students and alert instructor if a student record is missing. It will also stop the demonstration, which is displayed whenever a training session is not taking place.

8. SIGNOFF — Signoff terminates the current training session, closes files and activates the demonstration.

9. STIFLE — Stops speech validation.

In addition to these special keys, other functions are available to student and instructor using the standard typewriter keyboard. The instructor is able to freeze aircraft dynamics during a run, abort an exercise, request a mandatory replay or a hardcopy printout. He is also able to override any current task, thus activating the next sequential task, if phase 1 is completed. After the scoring of a P-run, he can modify the file to correct errors in speech understanding and rescore. When necessary, he may activate instructor functions at the trainee station. There are also several functions relating to the speech understanding. The instructor may initiate R/T practice to insure good voice reference patterns have been obtained. He can also request validation of new phrases and update of the phrase file. The trainee may also request R/T practice or ask for instructor assistance.

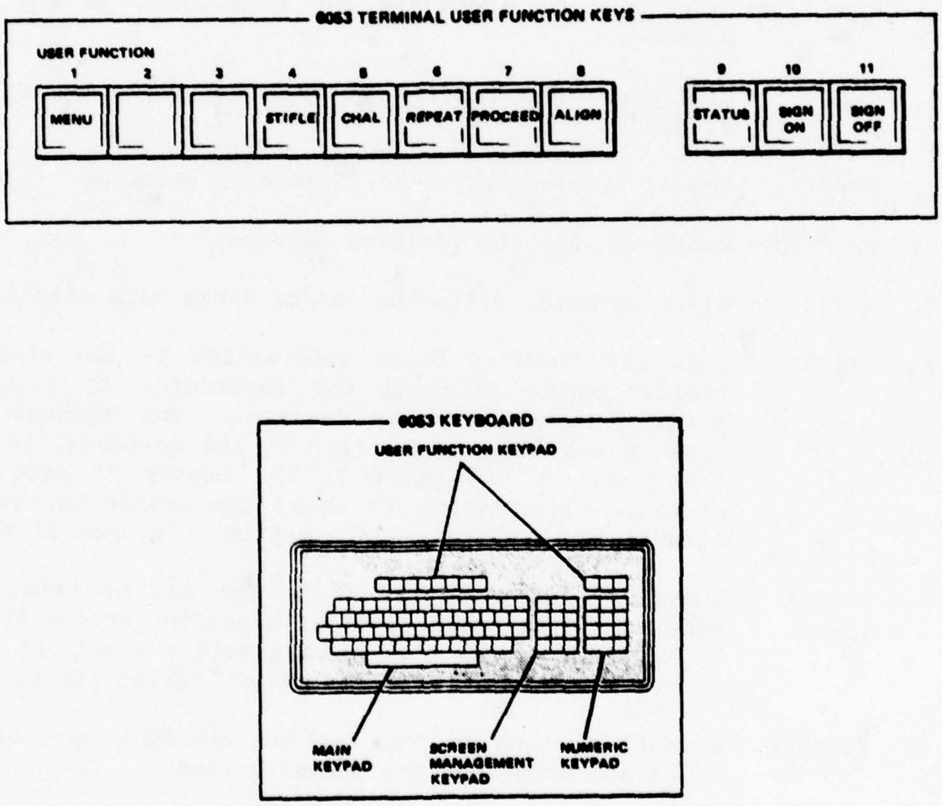


Figure 41. Student Keyboard

Any time a student or instructor asks for an invalid function the CRT echoes a question mark and waits for a valid entry. The list of valid functions is maintained by the subroutines SMENU (legal student inputs) and IMENO (legal instructor inputs).

Student and Instructor Panels Processing

Output processing for the lights and alarm on these communications panels is accomplished through PANOUT, a subroutine which can be called at any time to change the state of the panels. Inputs are acquired upon request or whenever a change occurs. The panel interrupt service routine activates a panel input processing routine, PIN. PIN has several duties. During replay it causes any panel inputs to be ignored. During phase 1 it handles the unusual rules which may be imposed such as blinking an ordinarily steady light to draw the student's attention to a particular button. During phases 2 and 3 it causes the panel to respond normally as defined in table 40. It maintains the state of the panels for use by applications routines, and also notifies some special routines when selected inputs are observed. It, like other routines, uses PANOUT to set the various lights and indicators.

TABLE 40. TRAINEE AND INSTRUCTOR PANEL CONDITIONS

Button	State	Light	Conditions
3	Off	Off	Pattern controller not in communication with GCA
	Off	Blinking Red	Pattern controller calls GCA
	On	Steady Red	Pattern controller and GCA communication link established
5,7			Operation the same as button 3, reserved for other controller positions
SUPER	Off	Off	Instructor not in communication with GCA
	Off	Blinking Red	Instructor calls GCA
	On	Steady Red	Instructor-GCA audio link established
ICS (instructor panel)	Off	Off	GCA not in communication with GCA
	Off	Blinking Red	GCA calls instructor
	On	Steady Red	GCA instructor audio link established
270.8	Off	Off	Radio frequency not in use
	Off	Amber	Radio frequency in use
	On	Amber, Alarm	Attempt to select radio frequency in use
	On	Green	GCA has use of radio frequency
318.8			As for 270.8
Foot Key	Keyed		Transmitting over selected radio frequency
	Unkeyed.		Not transmitting

TABLE 40. TRAINEE AND INSTRUCTOR PANEL CONDITIONS (CONT)

Button	State	Light	Conditions
MONITOR	Off	Off	Any communications over the radio frequency cannot be heard
	On	Amber	Communications on the associated radio frequency can be heard only
REQUEST	Off	Off	No request
	On (momentary)	Green	Clearance request posted to tower
CLEARED (light only)		Off	No clearance
		White	Landing clearance granted by tower
W/O		Off	No waveoff
		Red flashing and alarm	Waveoff ordered by tower
	On (momentary)	Off	Waveoff acknowledged by trainee

VOTRAX I/O

The Votrax uses a special package of subroutines and tasks to make audio output smooth and more natural. This package, called VSCON consists of the following programs:

1. VSCON — An assembler code FORTRAN callable subroutine. This routine is capable of processing up to 31 arguments, each of which relates to one phrase.

2. VSOUT — A FORTRAN task which processes the queued arguments by multi-buffering from FRAZ.VO to $VRO.

3. RDFRAZ — An assembler code task to read the voice unit phoneme file. Creates a phoneme stack.

4. WRFRAZ — An assembler code routine which reads the phoneme stack created by RDFRAZ and sends the information to $VRO.

The package also requires that a file called FRAZ.VO be created, containing the phrases to be output. Two common blocks, VSCOM.CO and VSFLAG.CO, as well as a symbol table for the phoneme stack, FRAZSYM, are also needed.

Record/Playback I/O

The speech digitizer is used both to collect speech for storage in digital format on the disk and to replay that speech. While speech is being collected, a buffer dump routine, SPDMP, writes it to the disk. SPOUT is a general purpose speech output routine which initiates a buffer filling task, then outputs the requested data.

SPOUT

Description: SPOUT controls speech digitizer output. It provides for stopping the digitized speech output at the end of an utterance so that the replay can be annotated. It is synchronized by an XMT from the caller through BXSPH.

Entry point: SPOUT

Classification: Task

Period: None

Language: A

Activated/called by: All routines requiring speech digitizer output

Cancelled by: SPOUT

Activates/calls: SPBUF

Cancels: None

Input arguments: CHNL — Channel on which speech data file is open

Common variables: RPSBF, RPSP1, RPSP2, NCDV, BXSPH, RPTSP, RPTDLY, BXSKY

Files created/changed: None

Files referenced: RPLSPH

PROCEDURE

FILL BUFFER A

IS ONLY PART OF BUFFER A TO BE OUTPUT?

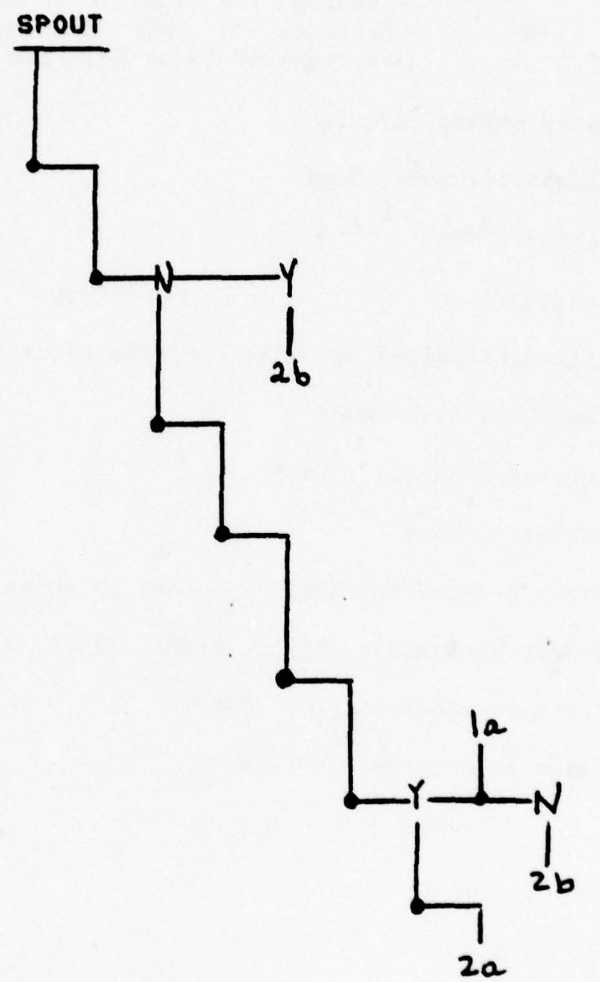
FILL BUFFER B

WAIT FOR XMT TO SYNCHRONIZE SPEECH OUTPUT (BXSPH)

START OUTPUT TO DIGITIZER

ARE MORE BUFFERS NEEDED?

WAIT FOR AN INTERRUPT INDICATING BUFFER OUTPUT COMPLETE (BXSKY)



SPOUT FLOWCHART (SHEET 1 OF 2)

SET CURRENT BUFFER TO EMPTY

SHOW NEXT BUFFER AS CURRENT BUFFER

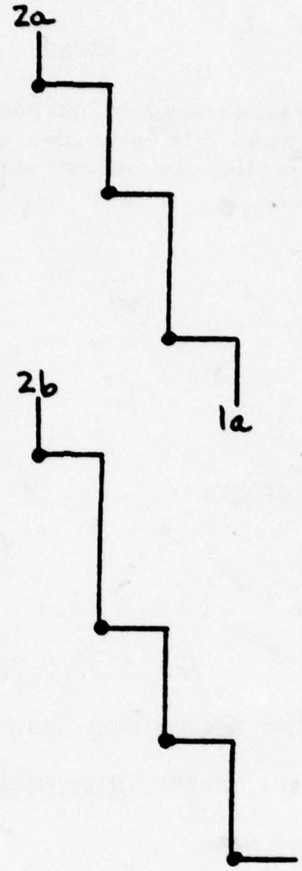
START SBUF

CLEAR BUSY, INFORMING SPEECH OUTPUT TO STOP AT THE END OF CURRENT BUFFER

DELAY .5 SECOND

WAKEUP EVSPT

DIE



SPOUT FLOWCHART (SHEET 2 OF 2)

SPBUF

Description: SPBUF fills speech output buffers with digitized speech information. It provides for stopping the digitized speech output at the end of an utterance so that the replay can be annotated.

Entry point: SPBUF

Classification: Task

Period: None

Language: F

Activated/called by: SPOUT

Cancelled by: SPBUF

Activates/calls: None

Cancels: None

Input arguments: CHNL — Speech data file channel

Common variables: RPSP1, RPSP2, NCDV, SDBF1, SDBF2

Files created/changed: None

Files referenced: RPLSPH

PROCEDURE SPBUF

IS HOLDING BUFFER FULL?

MOVE DATA FROM HOLDING
BUFFER INTO BUFFER A,
ZEROING PART WHICH
CORRESPONDS TO THAT ALREADY
OUTPUT

ARE ONLY PART OF THESE DATA
TO BE SPOKEN?

SET BUFFER A FULL

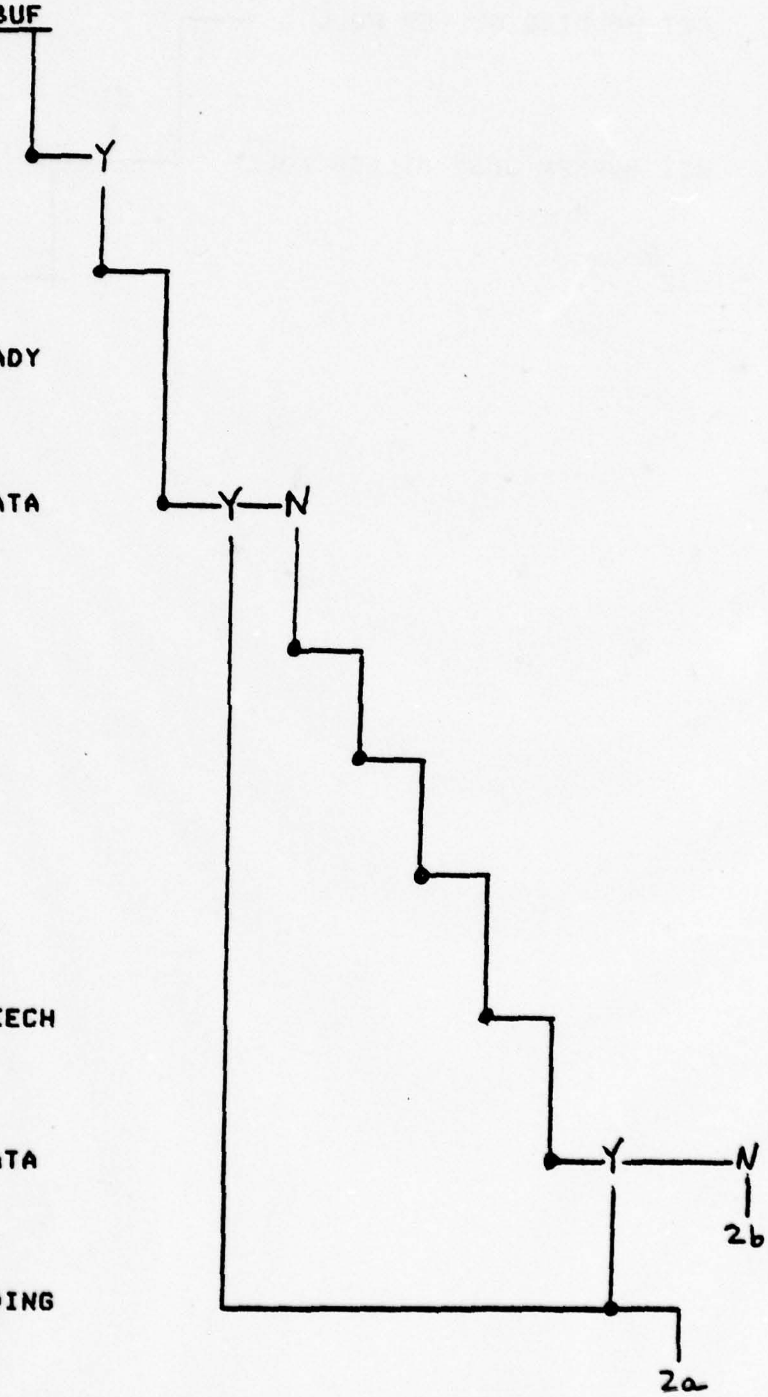
RELEASE HOLDING BUFFER

SET NEXT BUFFER TO B

FILL NEXT BUFFER WITH SPEECH
DATA

ARE ONLY PART OF THESE DATA
TO BE SPOKEN?

MOVE REST OF DATA TO HOLDING
BUFFER, ZEROING
CORRESPONDING PORTION OF
CORE



SPBUF FLOWCHART (SHEET 1 OF 2)

2a

SET HOLDING BUFFER FULL

2b

SET BUFFER JUST FILLED FULL

DIE

SPBUF FLOWCHART (SHEET 2 OF 2)

SPDMP

Description: This routine is activated by an .IXMT from the record/playback interrupt service routine when a buffer is full of digitized speech data. This routine will dump the buffer to the disk and release the buffer for further input.

Entry point: SPDMP

Classification: Task

Period: None

Language: A

Activated/called by: SPDMP

Cancelled by: P3TRM

Activates/calls: None

Cancels: None

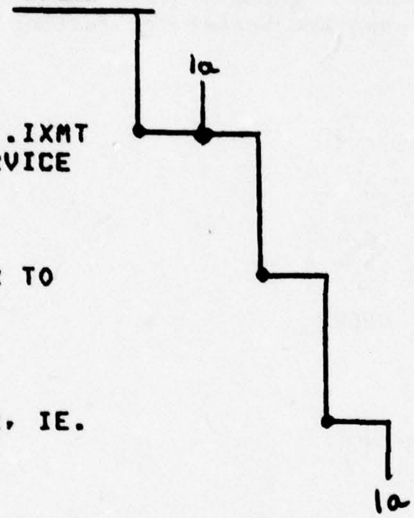
Input arguments: None

Common variables: NCDV, SDBF1, SDBF2, RPSP1, RPSP2, BXRC

Files created/changed: RPLSPH

Files referenced: None

PROCEDURE SPDMP



CALL REC, WAITING FOR .IXMT
FROM RPB INTERRUPT SERVICE

WRITE INDICATED BUFFER TO
REPLAY FILE

RELEASE LOCK ON BUFFER, IE.
INDICATE BUFFER EMPTY

SPDMP FLOWCHART (SHEET 1 OF 1)

Activity File Output

ACTOUT is a centralized routine which is called by all routines which wish to output data to the activity file. It provides the interface which enables scoring to proceed in real time during phase 2 and during replay in phase 3. It does this by filling the activity buffer directly in phase 2 for RDACT's use. In phase 3 it writes the activity file which is later read into the activity buffer.

ACTOUT

Description: This subroutine either writes data to the activity replay file in phase 3 or informs PMS in phase 2.

Entry point: ACTOUT

Classification: Subroutine

Period: None

Language: A

Activated/called by: All routines which require records in the activity file

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: I1,....I8 — the 8 words to be written

Common variables: BXRAT

Files created/changed: RPLACT

Files referenced: None

PROCEDURE

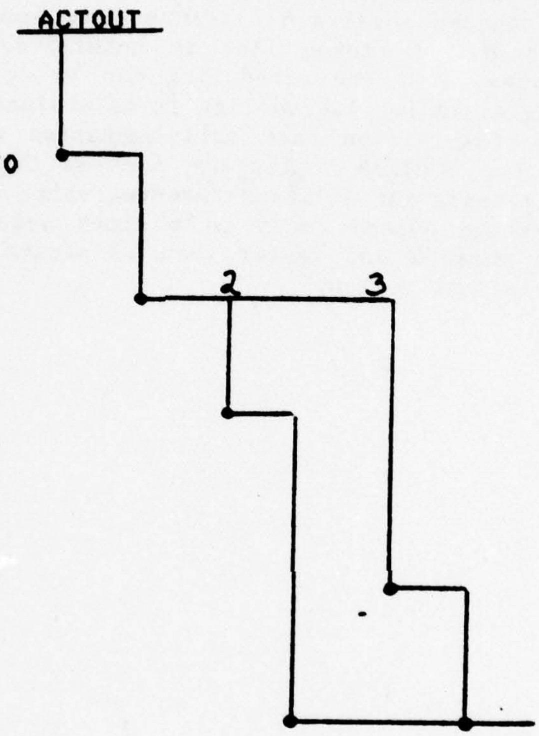
COPY INPUT ARGUMENTS INTO
BUFFER AREA

BRANCH ON PHASE

XMT ADDRESS OF BUFFER
CONTAINING DATA TO RDACT

START WRITE TASK TO
UPDATE ACTIVITY FILE

RETURN



ACTOUT FLOWCHART (SHEET 1 OF 1)

Range and Time-Related Task Queuing

These features are provided to allow the performance measurement routines and others to schedule required checks and initiate special processing when the aircraft reaches a certain distance from touchdown or when a period of time has elapsed. A linked list is maintained to order task and range entries by range, thus the scheduling can be requested dynamically as the run proceeds. A similar linked list is maintained for the time related task calls. This time-related task call mechanism enables PMS to score time outs, etc. The FORTRAN 5 library routine CYCLE would not satisfy the GCA-CTS requirements for a faster-than-real-time scoring capability. TIMCAL on the other hand depends only on a clock which runs in real .5 second increments in phase 2 and faster than .5 second increments during replay. Program descriptions follow.

RNGSCHD

Description: This routine maintains the linked list of tasks which are to be scheduled when the aircraft reaches a specified range from touchdown.

Entry point: RNGSCHD

Classification: Subroutine

Period: None

Language: A

Activated/called by: Any routine which schedules tasks based on range

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: NTREE — task LOCO entry
RANGE — Range at which task is to be scheduled
MSG — 1 word message to be passed to task

Common variables: SKREN, SKRNG, SKRAV

Files created/changed: None

Files referenced: None

PROCEDURE RNGSCHD

ADD THIS TASK TO THE LIST OF
THOSE TO BE SCHEDULED

UPDATE LINKS SUCH THAT RANGE
ORDER IS PRESERVED

RETURN

RNGSCHD FLOWCHART (SHEET 1 OF 1)

RNGCAL

Description: RNGCAL activates tasks on the basis of the aircraft's distance to touchdown, and provides a one word message.

Entry point: RNGCAL

Classification: Subroutine

Period: .5 second

Language: F

Activated/called by: APE

Cancelled by: N/A

Activates/calls: Any routine which requested activation at a specific range by means of a call to RNGSCHD, TIMCAL

Cancels: None

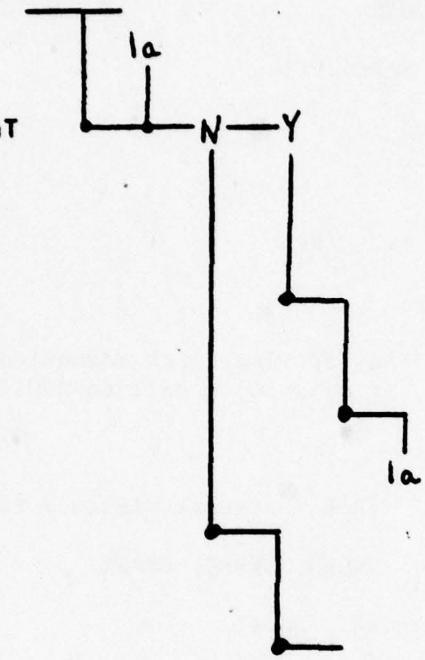
Input arguments: RANGE — present distance to touchdown

Common variables: SKREN, SKRNG, SKRNX

Files created/changed: None

Files referenced: None

PROCEDURE RNGCAL



IS THE INPUT RANGE < = THAT ASSOCIATED WITH THE NEXT TASK TO BE ACTIVATED?

ACTIVATE TASK

UPDATE POINTER INTO QUEUE

START TIMCAL

DIE

RNGCAL FLOWCHART (SHEET 1 OF 1)

TIMSCHD

Description: This routine is used to schedule tasks after an interval of time. It maintains a linked list of tasks which are to be started.

Entry point: TIMSCHD

Classification: Subroutine

Period: None

Language: A

Activated/called by: PMS routines

Cancelled by: N/A

Activates/calls: None

Cancels: None

Input arguments: NTREE — task LOCO entry
TIME — time according to CLTICK when task is to be started

Common variables: TNTREE, TAVAIL, TNXT

Files created/changed: None

Files referenced: None

TIMCAL

Description: TIMCAL activates tasks on the basis of the time according to CLTIME.

Entry point: TIMCAL

Classification: Task

Period: .5 second during phase 2, after every activity record in replay

Language: F

Activated/called by: RNGCAL

Cancelled by: P3TRM

Activates/calls: Tasks in TNTREE table

Cancels: None

Input arguments: None

Common variables: CLTIME, TNTREE, TNXT

Files created/changed: None

Files referenced: None

Random Number Generator

A random number generator is required to select varying environments in which the controller is expected to operate. The routine to be used in GCA-CTS is an implementation of an algorithm referenced in The Art of Computer Programming, Volume 2, by Donald E. Knuth. The routine is a FORTRAN 5 integer function. The function is passed an integer seed and returns a pseudo-random integer in the range $0 \leq N \leq 2^{16} -1$. If a function between 0 and 1 is desired, the number can be obtained by the computation

$$F = N/2^{16}$$

where N is the integer number originally returned by RANDOM.

User Clocks

Several user clocks are defined during phase 1 training to implement various "wait" instruction and voice data collection related tasks. None of these clocks need be set upon phase 1 initiation. Each clock is defined and removed as a particular timeout period is defined.

The "wait" instructions incorporated in the phase 1 task files specify timeout periods for given "wait" conditions; e.g., wait for student keyboard entry. For this condition, a clock is defined for the length of time allotted for a keyboard entry (refer to P1WAT). If a keyboard entry does not occur before the timeout, the specified task file branch is taken.

Voice data collection related clocks are basically used to time the student's lack of input. If timeouts occur, voice data collection routines attempt to discern the presence or absence of the trainee. In some instances another prompt is issued.

Replay depends upon a user clock to recreate the timing of the original run and to ensure the synchronization of replayed speech data and feedback messages.

Device Drivers

The GCA-CTS uses several non-standard devices. Each requires a device driver tailored to its own peculiarities. Brief descriptions of the Threshold 500, Votrax, input panels and record/playback device drivers are given in the following paragraphs. Graphics display and joystick I/O was described in Section 4.

Threshold 500 Device Driver

The voice input preprocessor, the Threshold 500, has a driver called VIPDR. VIPDR performs three functions: turns the unit on (VIPON), turns it off (VIPOFF), and services interrupts (VINTS/VINPT).

VIPON enables voice input processing. The device is identified to RDOS, cleared, and attached to the RDOS interrupt structure. Input buffer flags are also cleared for voice input deposits.

VIPOFF disables voice input processing. The device is removed from the RDOS interrupt structure and device list.

VINTS/VINPT services interrupts. VINTS services the first interrupt as a voiced phrase is input. A free input buffer is selected for incoming voice data storage. VINPT then fills the selected buffer. Interrupts which follow report directly to VINPT. An interrupt from the preprocessor should occur every two milliseconds until a long pause (silence of 0.1 second duration) is detected. Upon detection of a long pause, the voice data are tagged and a message is sent to announce the data input.

Votrax Speech Synthesizer Driver (VRODR)

The Votrax is an operating system device ($VRO) consisting of one root code module and two overlay modules. The device operates in two modes. Write sequential (.WRS) transfers octal codes directly to $VRO. Write line (.WRL) translates ASCII phoneme format to octal codes and then outputs. Generally the vocabulary required is translated to octal codes so that .WRS can be used. In this way the data are more compressed and therefore retrieval and output are faster. Implementation as an operating system device makes it possible to develop and modify vocabulary files easily with EDIT.

Student/Instructor Panels

The inputs from both panels are considered to be from one device as described in Section 3. The RDOS .IDEF system call is used to define this device to the operating system at runtime. Its interrupt service routine (SIPDR) gains control when a panel change-generated interrupt is received. Two words of data are retrieved with a DIA and DIB, and the processing of these data is initiated by an .IXMT to PIN.

Record/Playback Unit

The record/playback device driver (RPBDR) detects the buffer full interrupts and initiates buffer storage while recording is taking place. During playback, it detects the buffer empty interrupts and initiates filling them with recorded speech data. The data channel device is introduced to RDOS at runtime through the use of the .ID EF s stem call.

APPENDIX A

Trainer Facilities Report

General Information

This facilities report is submitted in accordance with UDI-H-2112 and describes the necessary data to define facilities requirements at the Naval Air Technical Training Center (NATTC), Memphis, TN. Specification of power and air conditioning requirements is also applicable to the temporary installation of the system at the Naval Training Equipment Center, Orlando, FL. Details are based on agreements made on May 23, 1978 in a conference and site inspection at NATTC, Memphis.

Site Preparation

Figure A-1 presents a plan view of the installation. There are three main equipments to be installed in fixed positions as noted. In addition, there are several movable items. The positions shown are dictated largely by space available rather than by operational requirements. The main operational requirement is that cable runs be less than 100 feet between either station and the computer cabinet.

The three fixed positions shown in figure A-1 are all on an elevated floor and all are adjacent to existing connecting cable trays below that floor. No units require special cooling ducts.

Site preparation details are presented in the following sections.

The System Controller

Size and Weight. The cabinet is a double bay cabinet with dimensions: 46" wide x 32" deep x 70" high. It is supported on six caster wheels and weighs approximately 1000 pounds.

Maintenance Access. Some units in the cabinet slide forward for maintenance. Access to cables is through double doors in the rear. Access space extending four feet to the front, four feet to the rear and two feet to the right is needed for maintenance. The space to the right is needed only for board removal from a CPU extended on its slides.

Power Requirement. All cabinet power enters the rear of the cabinet through an eight foot cable of 7/8-inch outside diameter. This cable is terminated at the supply end in a plug NEMA type 14-50R. The plug will mate with a NEMA 14-50P receptacle.

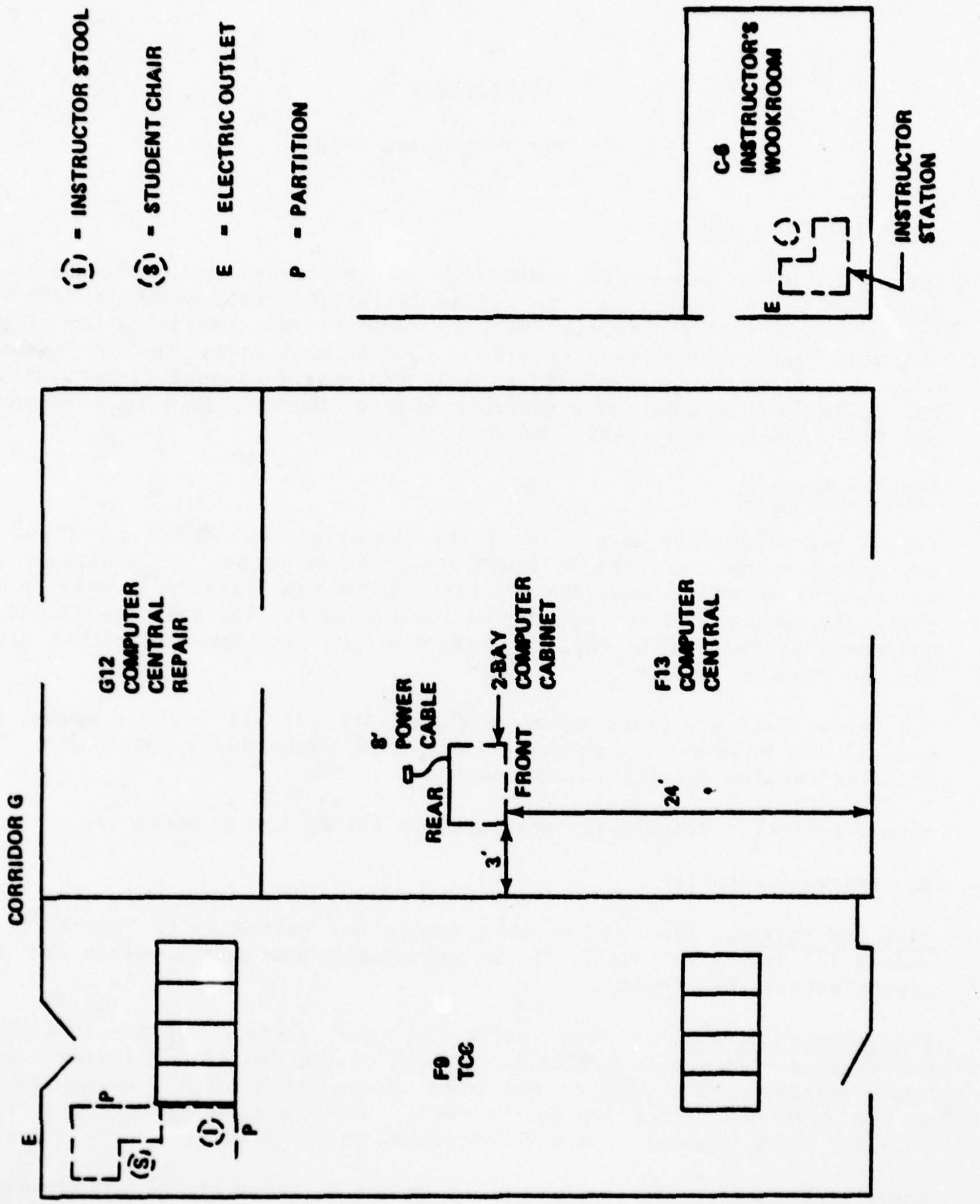


Figure A-1. Proposed GCA-CTS Equipment Locations for NATTC, Memphis

The receptacle shall supply two phases of 120 volt, 40 amperes separated by a phase angle of 120 or 180 degrees. In other words, the power source may be 120/240 volts or 120/208 volts, three phase (only two phases being wired through the receptacle). Within the cabinet only 120 volt service is used.

The voltages should be within + 5 percent of above values and the frequency should be 60 Hz. A safety ground independent from the electrical neutral shall be provided.

Figure A-2 indicates the receptacle. It may be mounted to the floor or subfloor or may be on a pigtail within reach of the cabinet's eight foot cable.

Air Conditioning. The cabinet has internal fans at the bottom front which pull in room air. Air is exhausted from the top of the cabinet. The cabinet will generate a maximum of 12,000 BTU per hour, thus requiring a maximum of one ton of air conditioning from the room cooling system.

Ambient intake air is assumed to be below 80°F.

Grounding. No grounding other than that provided by the power receptacle is needed.

Floor Cutouts and Loading. A square floor cutout at least six inches on a side is needed just behind the cabinet (centered) to allow cabling to pass to the subfloor trays.

If the power receptacle is below the elevated floor, the above hole should be at least eight inches on a side to allow for the power plug. Alternately, a separate power cable hole may be provided to clear a power plug of 3 1/2 inches in diameter.

All holes should have rounded, non-abrasive edges.

Floor loading totaling 1000 pounds is concentrated on four casters located near the corners and two casters near the front and back center.

Other Considerations. The intended installation area offers adequate illumination. Existing acoustic noise and vibration levels pose no problems. There should be no problem with generation of or susceptability to electromagnetic radiation.

The Trainee Station

Size and Weight. The station consists of a desk with several equipment items on the top surface. See figure A-3.

With the equipment items in place, the overall dimensions are 72" wide x 74" deep x 50" high (to top of equipment). The station will be assembled in its final position.

Weight will be approximately 800 pounds, distributed on the floor by six adjustable leveling feet.

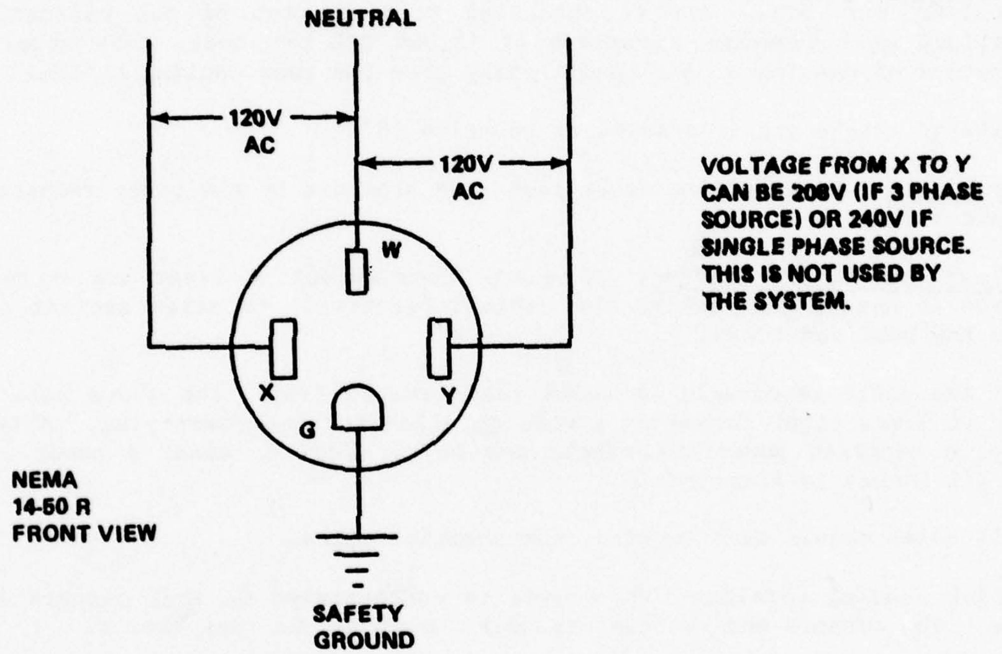


Figure A-2. Power Receptacle for System Controller

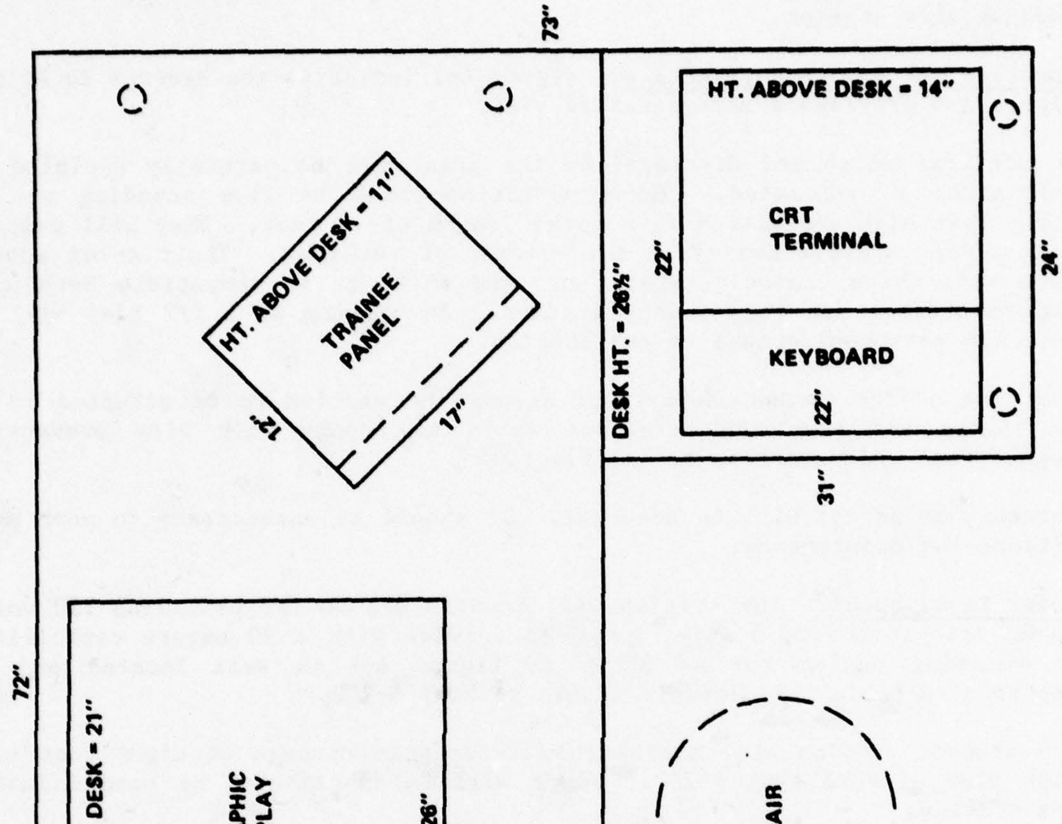


Figure A-3. Trainee Station Top View

A student's chair and an instructor's stool, as described later, will be used at this station.

Location and Maintenance Access. Figure A-1 indicates the general location. Figure A-4 provides a more detailed view.

To minimize noise and distractions the area will be partially enclosed by partitions as indicated. These partitions will be free standing six to eight feet high and will have a total length of 18 feet. They will completely divert traffic away from this corner of the room. Their color should be a soft white, colonial white, or sand white to be compatible both with the room decor and the student station. An opening of 2 1/2 feet will be left for personnel access to the station.

The size of the framed compartment allows the station to be situated 18 to 24 inches out from the permanent walls to comply with fire prevention regulations understood to be in effect.

Maintenance access will be adequate. It should be unnecessary to move partitions for maintenance.

Power Requirement. The station will require one outlet providing 120 volt, 60 Hz, single phase, 3 wire, grounded service with a 20 ampere capability. An existing outlet box as shown in figure A-1 is well located and is reported suitable. Receptacle should be NEMA 5-20R.

The student station will access this receptacle through an eight foot cord with plug of NEMA type 5-20P. Power will be distributed as needed inside the station.

Air Conditioning. Equipments in the student station will be cooled by room air through internal fans as required. The system will generate less than 6000 BTU per hour, thus placing one-half ton maximum load on the room air conditioner.

Ambient air is assumed to be below 80°F.

Grounding. No grounding other than that provided by the power receptacle is needed.

Floor Cutouts and Loading. A square floor cutout six to seven inches on a side is needed near the rear center of the student station. See figure A-4 for positioning. This will allow the signal cables to pass to a subfloor cable tray. The hole should have rounded, non-abrasive edges.

Floor loading totaling 800 pounds is concentrated on six adjustable leveling feet as indicated in figure A-2.

Acoustic Noise Level. Reasonably low noise levels are needed to permit good speech recognition. While exact noise characteristics and levels are hard to define, it appears that the broad-band audio acoustic noise level should

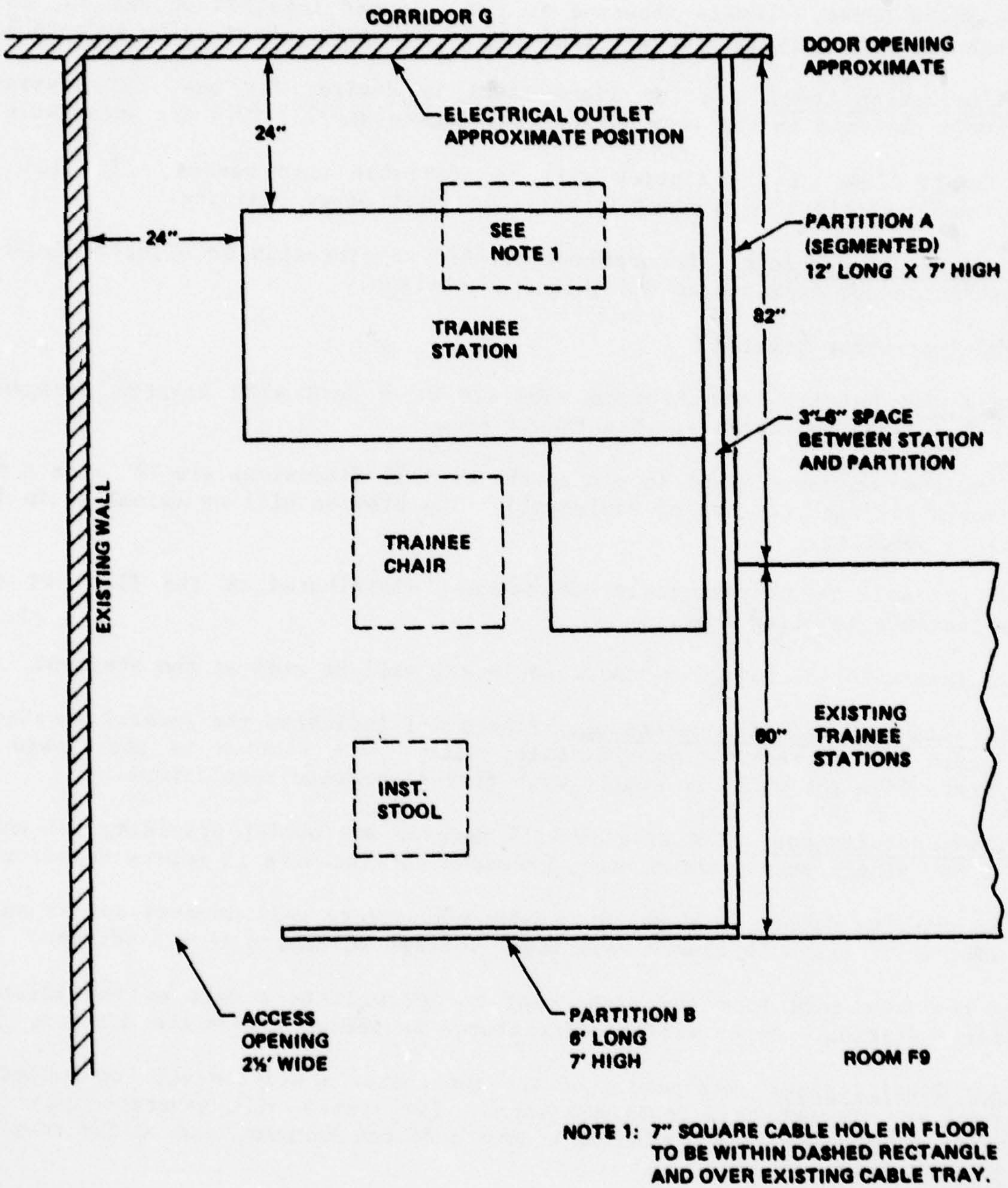


Figure A-4. Trainee Station Floor Plan

be below 50DBA. Levels observed at the intended location on May 23, 1978 with approximately 12 students training in the room appear quite acceptable.

Illumination Level. A low light level is desired for best CRT viewing. Levels observed in the room during training on May 23, 1978 are acceptable.

A small light at the station will be available when needed. It will be properly shielded to prevent interference with other training.

Other Considerations. No problems related to vibration or electro-magnetic vibration are expected at the intended location.

The Instructor Station

Size and Weight. The station consists of a desk with several equipment items on the top surface. See figure A-5.

With the equipment items in place, the overall dimensions are 72" wide x 65" deep x 44" high (to top of equipment). The station will be assembled in its final position.

Weight will be approximately 800 pounds, distributed on the floor by six adjustable leveling feet.

An instructor's chair, as described later, will be used at the station.

Location and Maintenance Access. Figure A-1 indicates the general location. Figure A-6 provides a more detailed view. The station is positioned 18 inches from the walls to comply with fire prevention regulations.

Power Requirement. The station will require one outlet providing 120 volt, 60 Hz, single phase, three wire, grounded service with 15 ampere capacity.

An existing outlet as shown in figure A-5 appears well located and is suitable if it has a 15 ampere capacity. Voltage tolerance is + 5 percent.

A ten foot cord from the right rear corner will be a part of the instructor's station. Power will be distributed as needed inside the station.

Air Conditioning. Equipments of the instructor's station will be cooled by room air through self-contained fans. The system will generate less than 4000 BTU per hour, thus placing a one-third ton maximum load on the room air conditioner.

Ambient air is assumed to be below 80°F.

Grounding. No grounding other than that provided by the power receptacle is needed.

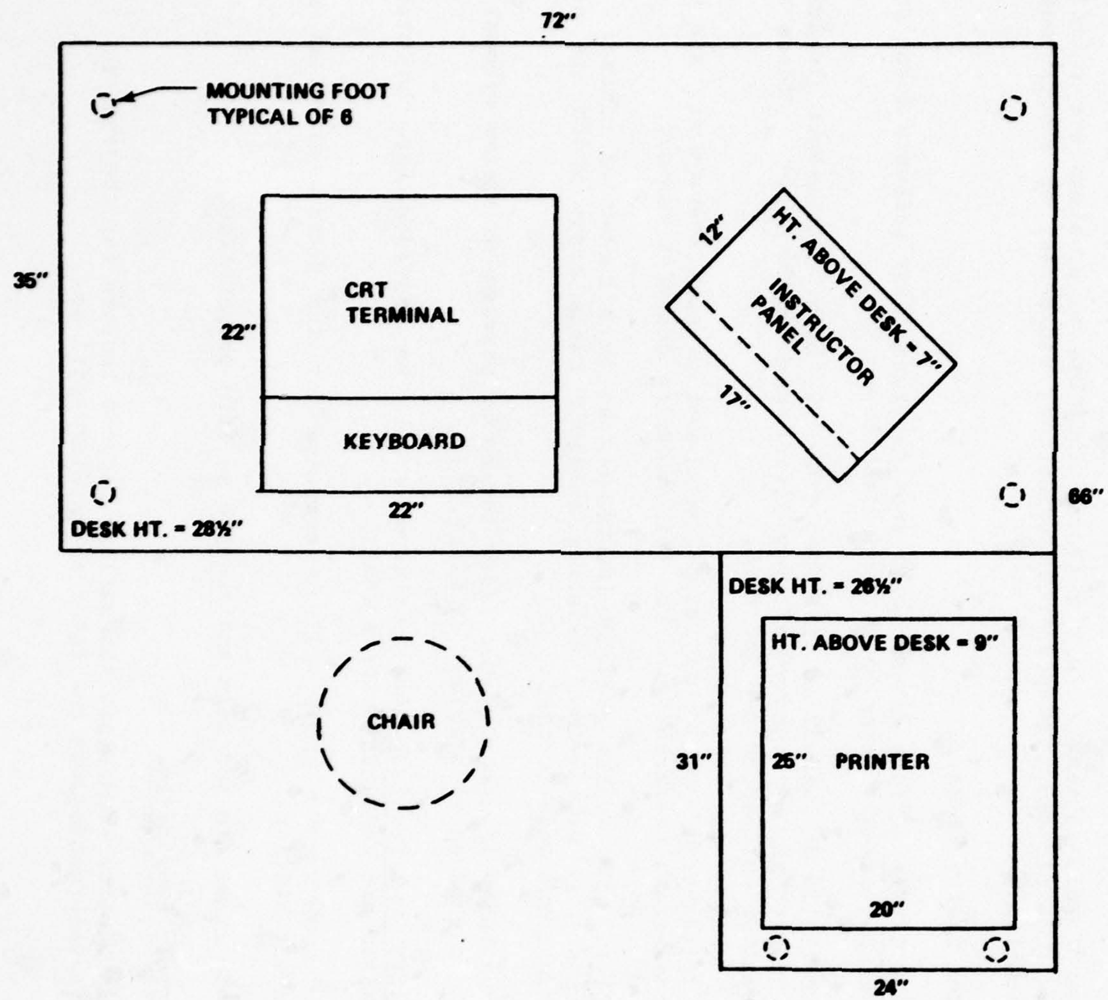


Figure A-5. Instructor Station Top View

Floor Cutouts and Loading. A square floor cutout six to seven inches on a side is needed near the rear center of the instructor station. See figure A-6. This will allow the signal cables to be passed to a subfloor cable tray. The hole should have rounded, non-abrasive edges.

Floor loading totaling 800 pounds is concentrated on six adjustable leveling feet.

Other Considerations. In the chosen location no problems are anticipated from acoustic noise, improper illumination, vibration, or electro-magnetic radiation.

Other System Items

Media Cabinet. This cabinet will provide storage for software items such as listings, manuals, disk packs, and diskettes.

Its size is 36" wide x 18 1/2" deep x 84 1/2" high. Its weight, loaded, may be as much as 600 pounds uniformly distributed on the floor. Access is via a roll-up curtain on one 36" side.

It requires no power. It should be placed in some convenient, air conditioned area; preferably in room F-13 near the computer cabinet.

Cables. The system will be interconnected by a number of cables. These cables will make use of existing subfloor cable trays which are already present and well located.

Three holes in the elevated floor for cable passage to system elements have been discussed in previous sections.

Chairs and Stools. One chair each will be provided for the student's station and the instructor's station.

One instructor's stool will be provided for the instructor's use at the student station.

These items are portable and require no site preparation.

Support Facilities

This section summarizes the various support factors for the system. Some of this information was presented in earlier sections.

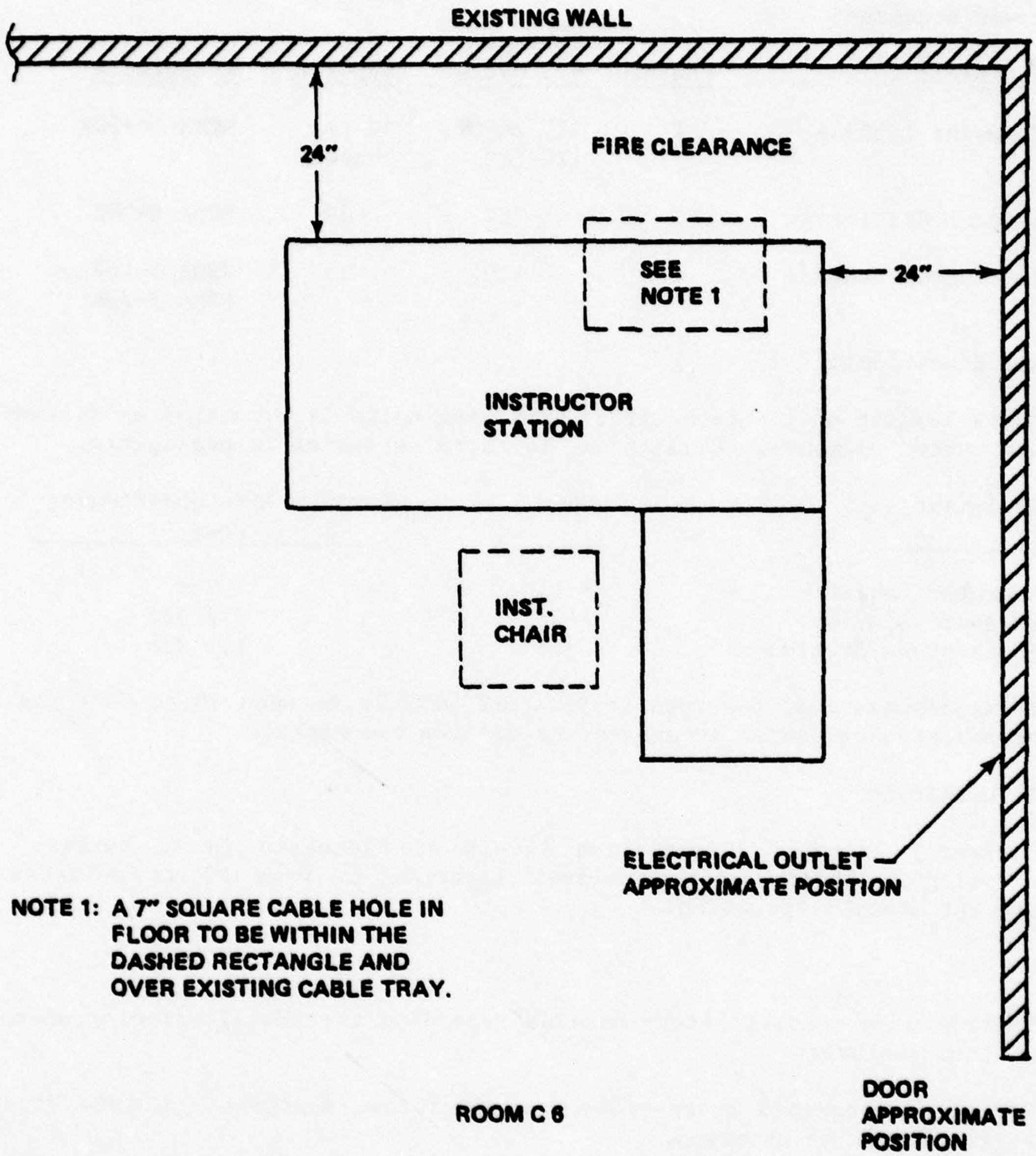


Figure A-6. Instructor Station Floor Plan

Power Required

Equipment	Phases	Voltage + 5 percent	Amperes	Receptacle
Computer Cabinet	2	120/240 or 120/208	40 per phase	NEMA 14-50R
Student Station	1	120	20	NEMA 5-20R
Instructor Station	1	120	15	NEMA 5-15R or NEMA 5-20R

Air Conditioning

Extra loading on the room air conditioning units is estimated as follows for the system elements. Loading due to extra personnel is negligible.

Equipment	Room	Maximum Air Conditioning Load
Computer Cabinet	F-13	1 Ton
Student Station	F-9	1/2 Ton
Instructor Station	C-6	1/3 Ton

It is assumed that the room temperature will be between 60 to 80°F and that no heaters are needed to exceed the minimum temperature.

Illumination

Currently existing illumination levels are adequate in all areas. The existing capability for low-level lighting in room F9 is suitable for student station operation.

Safety

There are no special safety hazards regarding the installation or operation of this equipment.

Normally recognized precautions for electrical equipment of this size and weight should be observed.

No special requirements for lightning protection exist for personnel or equipment.

Vibration and Acoustic Noise

The system generates a minimum of noise and vibration. Accordingly, its installation does not require any special footings, isolation, or other reduction techniques.

Electro-magnetic Radiation

The system consists almost entirely of commercially available hardware.

Radiation and susceptability are assumed to be low enough so as to permit coexistence with existing equipments.

Facilities Provided by Logicon

Logicon will assemble and test the system on site and will provide the test instruments and tools necessary to do so.

It is assumed that the site will be ready prior to Logicon's arrival. This includes having the space cleared, having necessary holes cut in the elevated floor, and having electrical receptacles available.

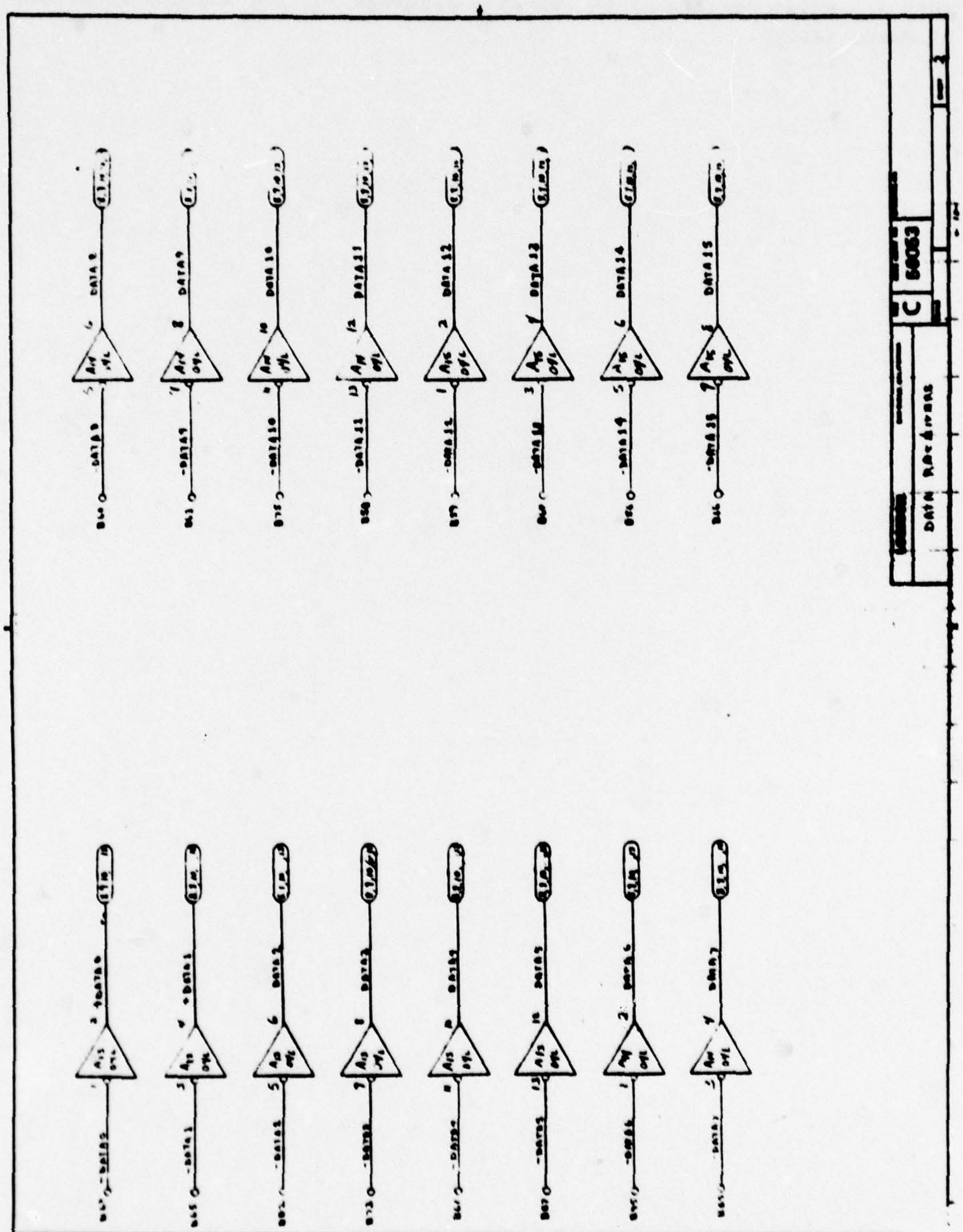
AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

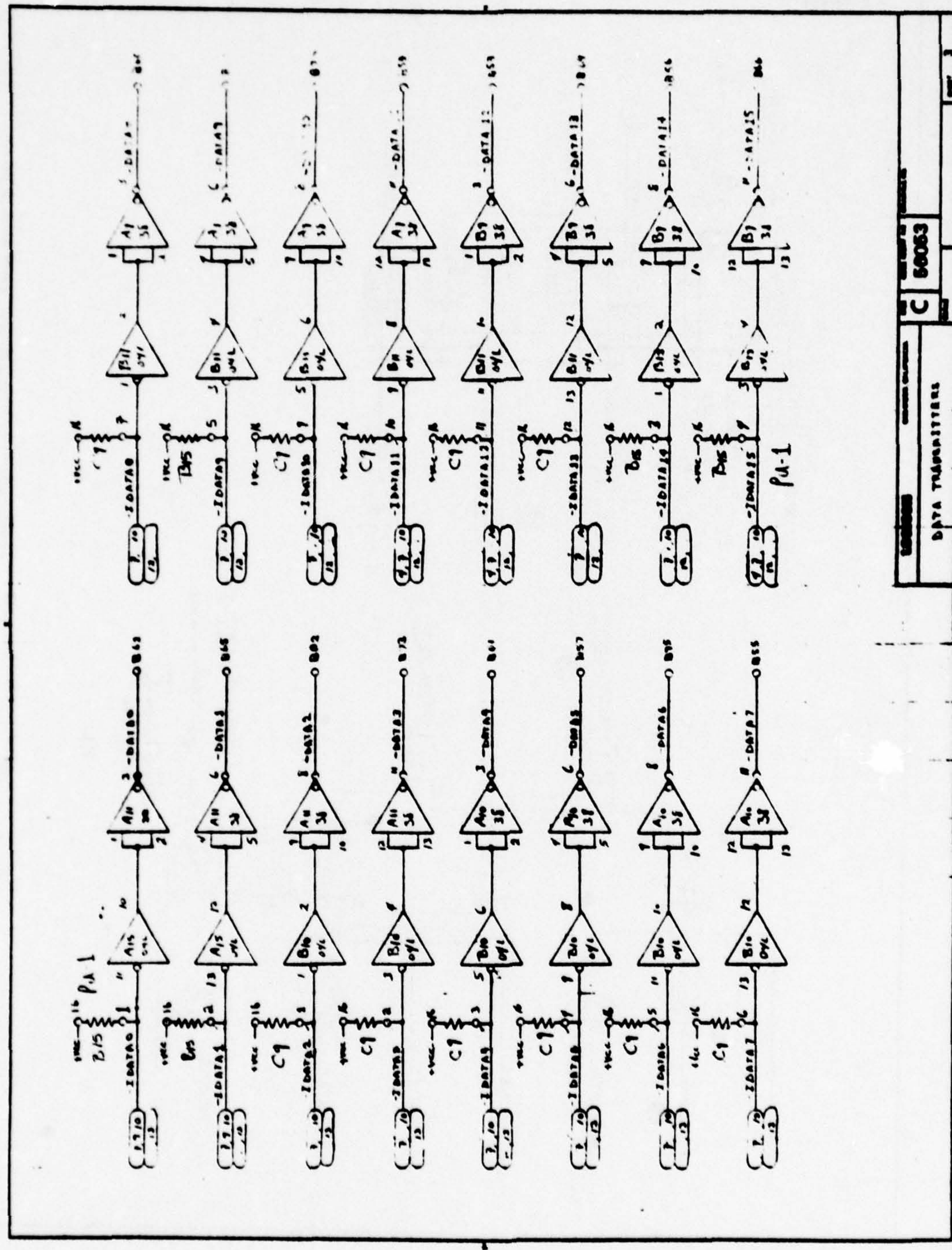
8 OF 9
AD
A069036

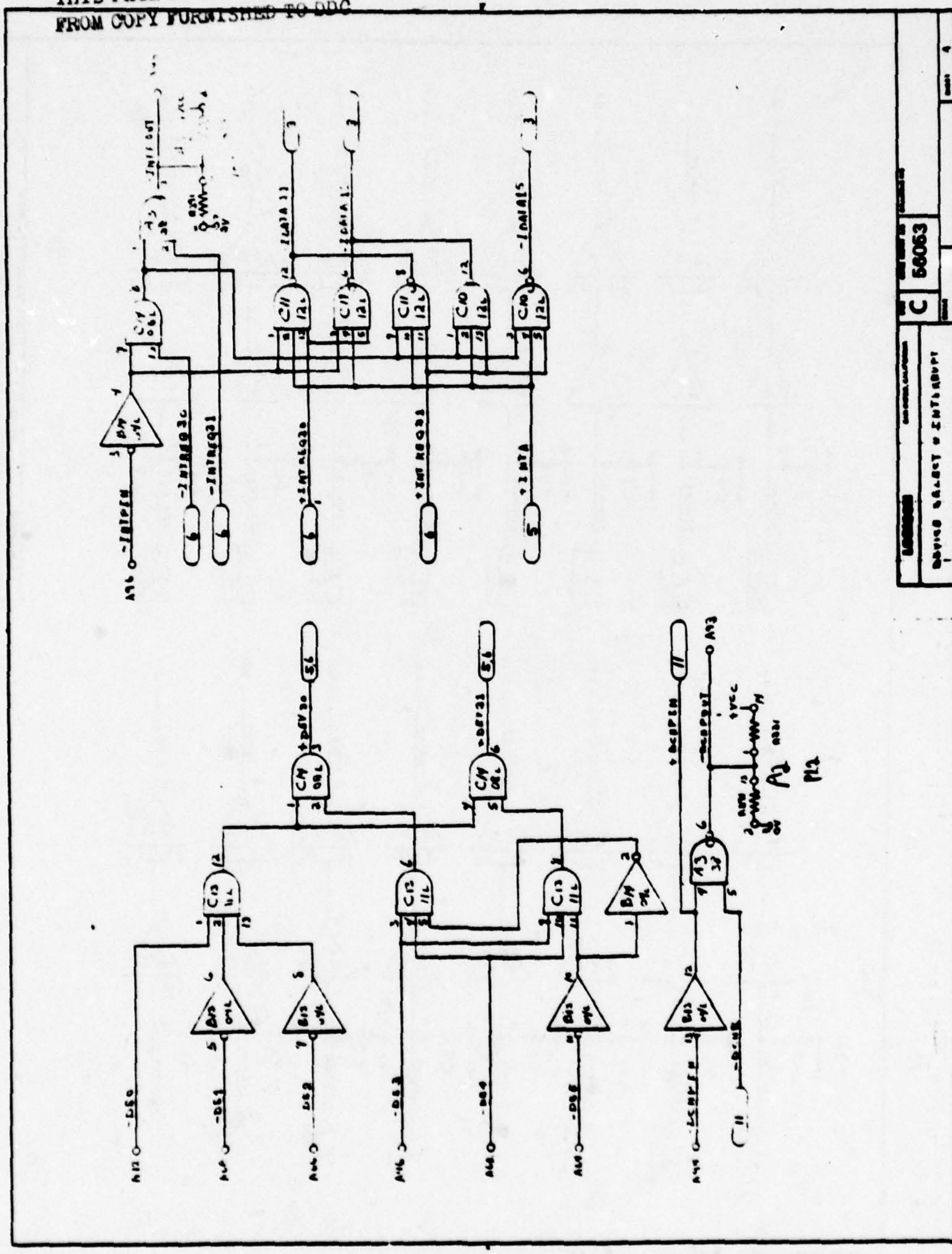
APPENDIX B

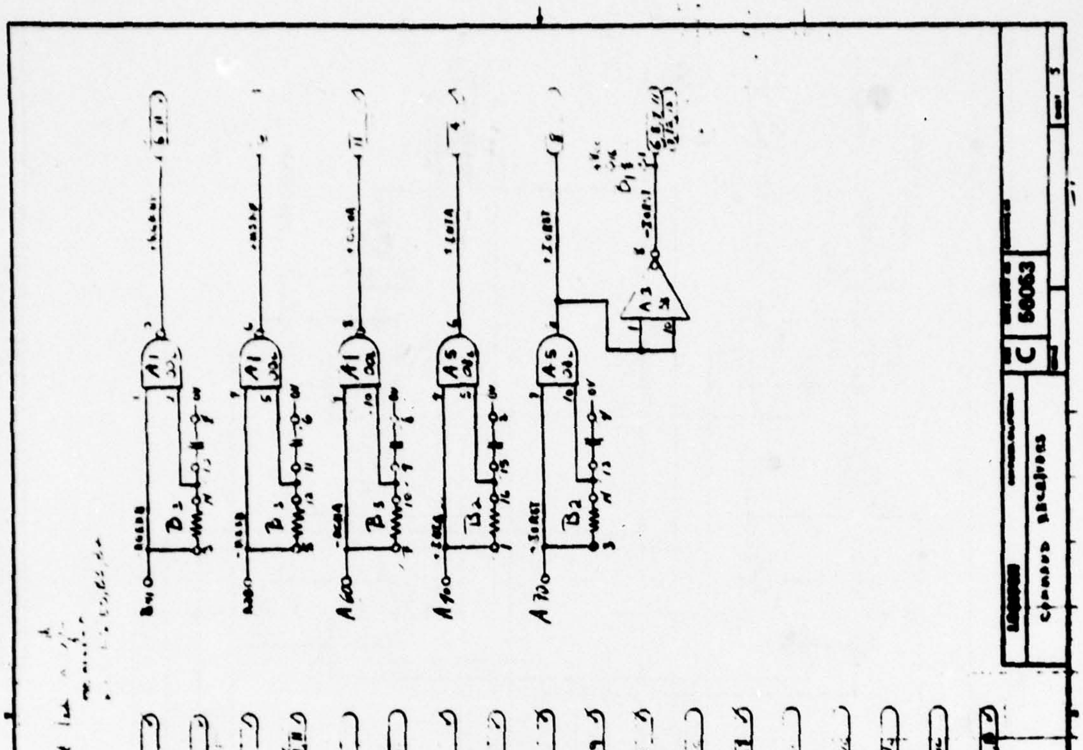
This appendix consists of the working drawings for the Logicon interface hardware design.

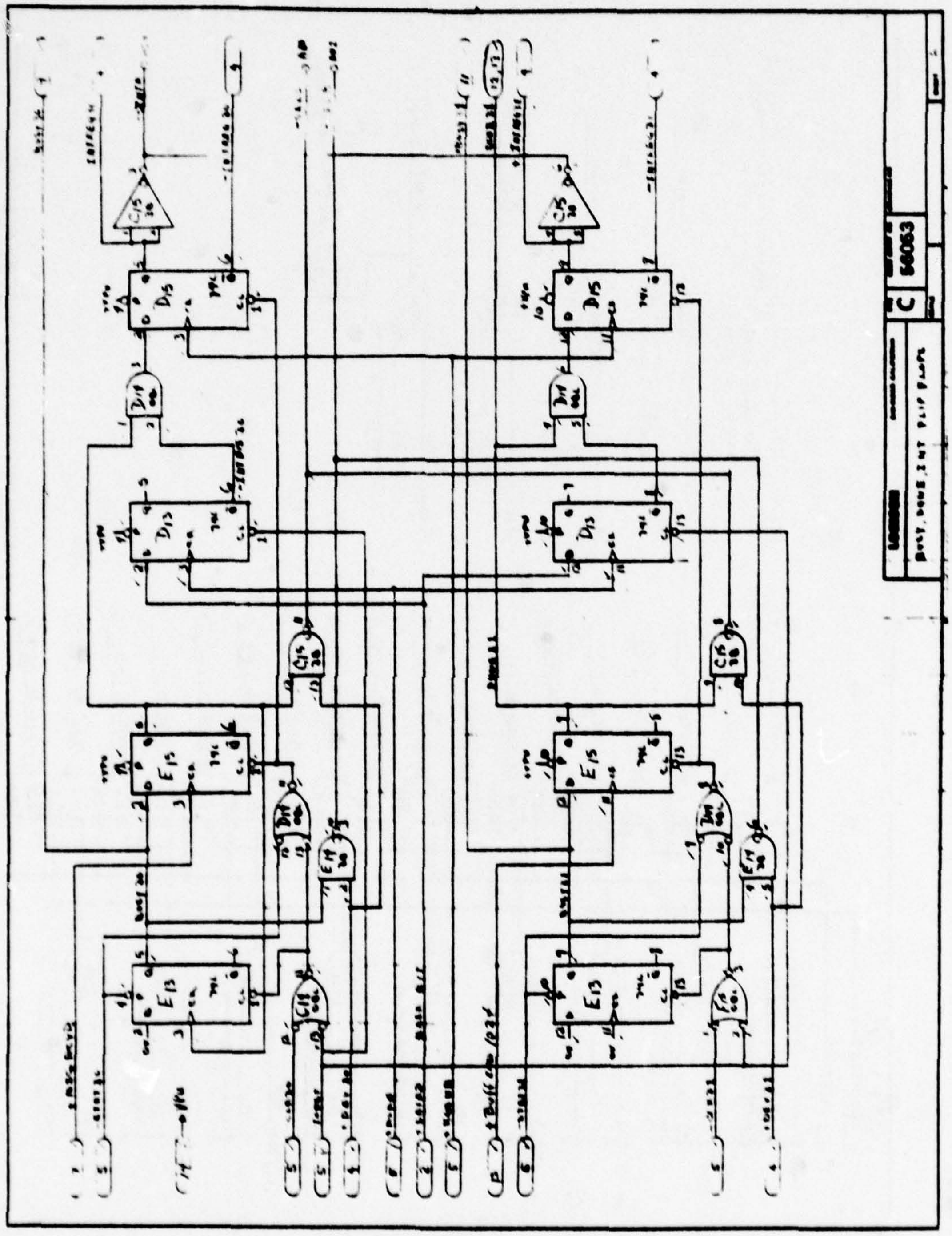
THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

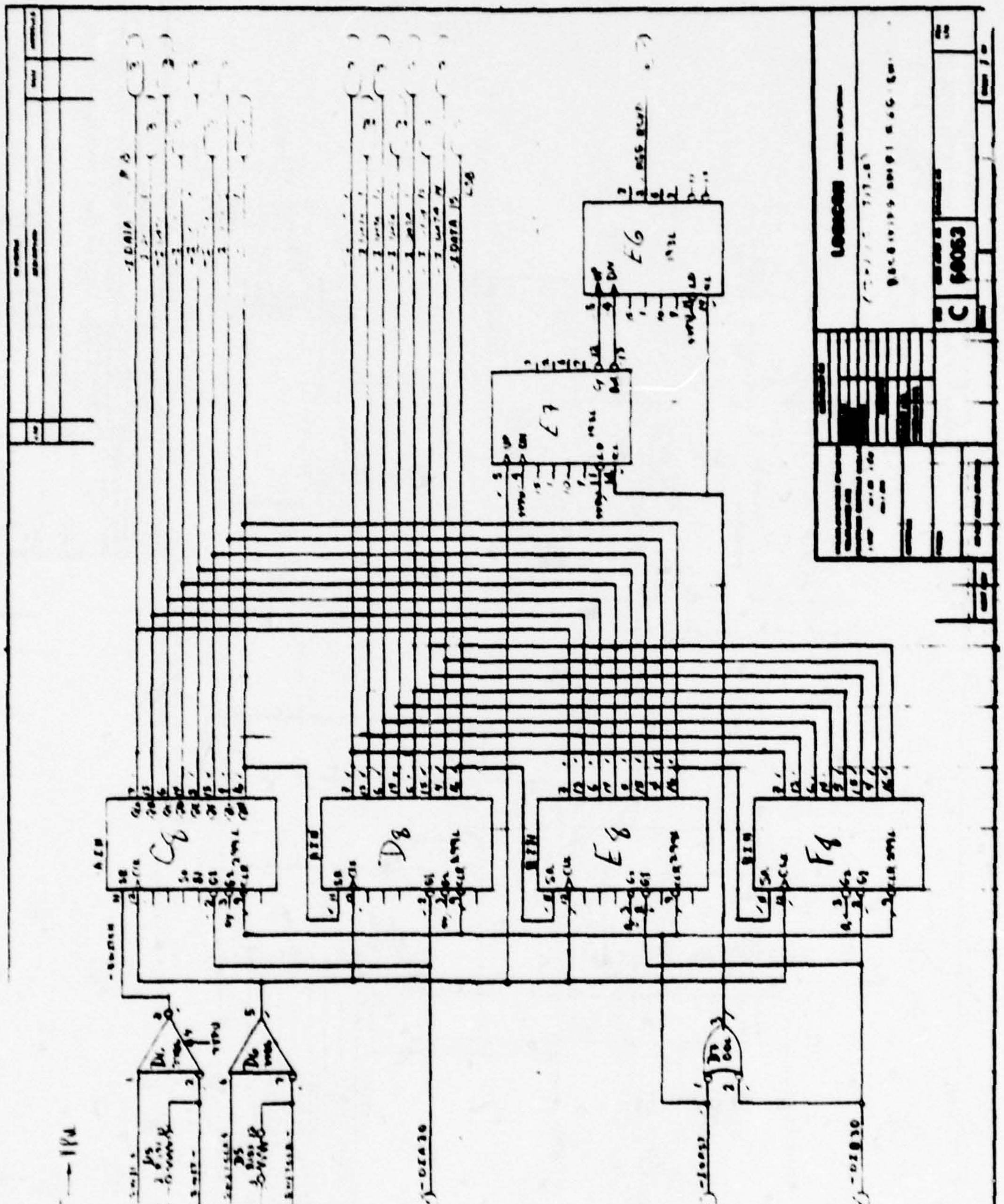


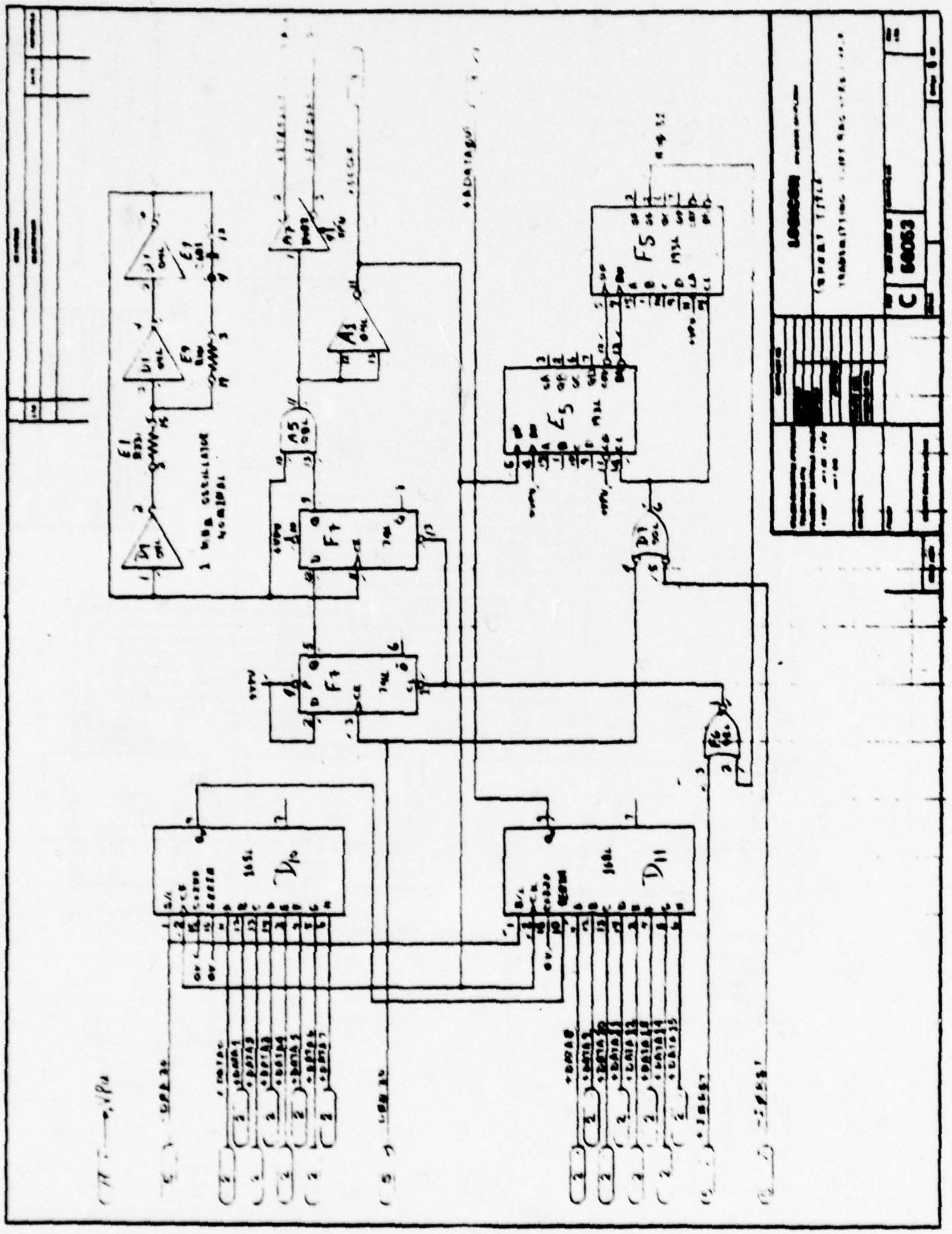

DATA TRANSMITTERS
C
56053

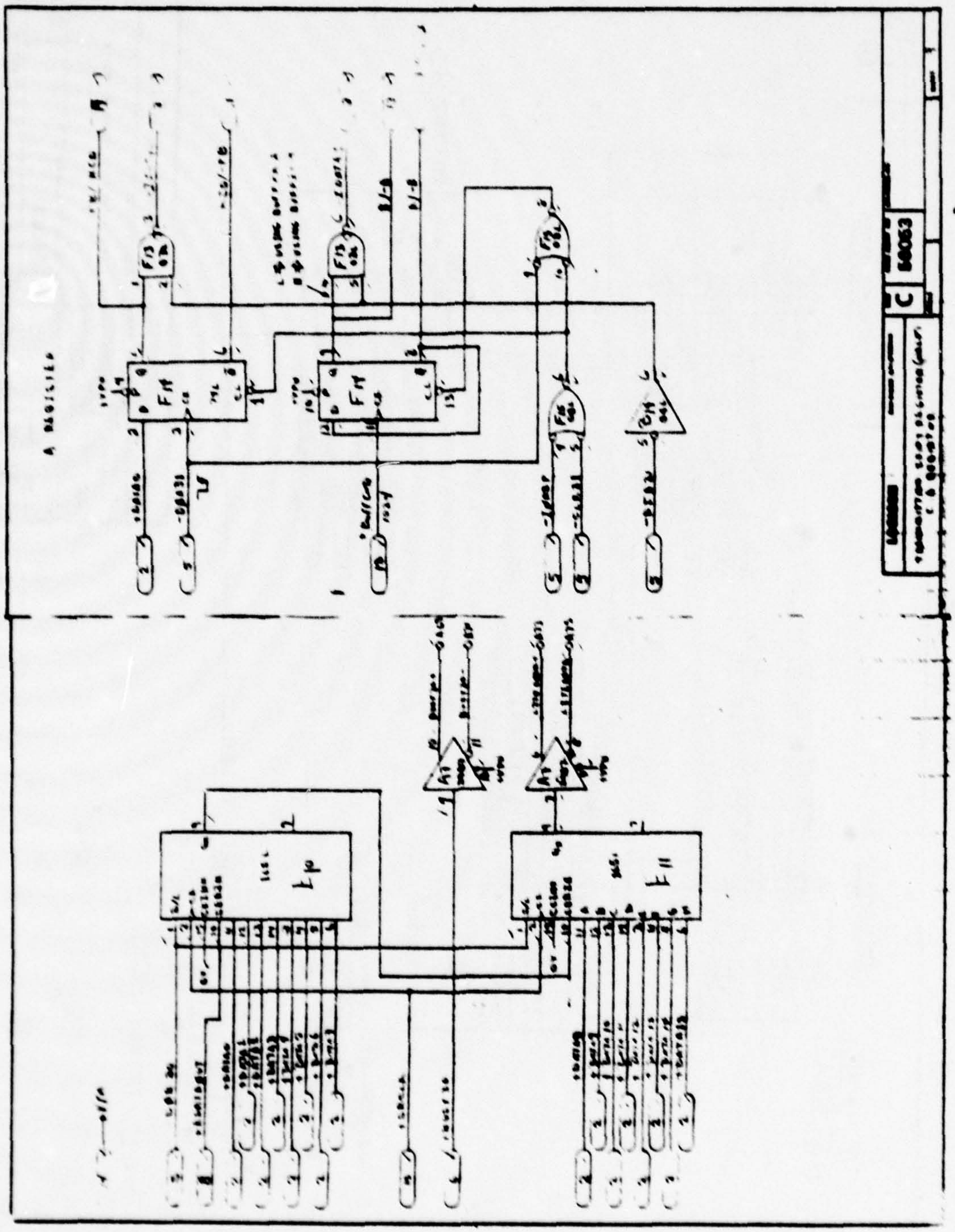


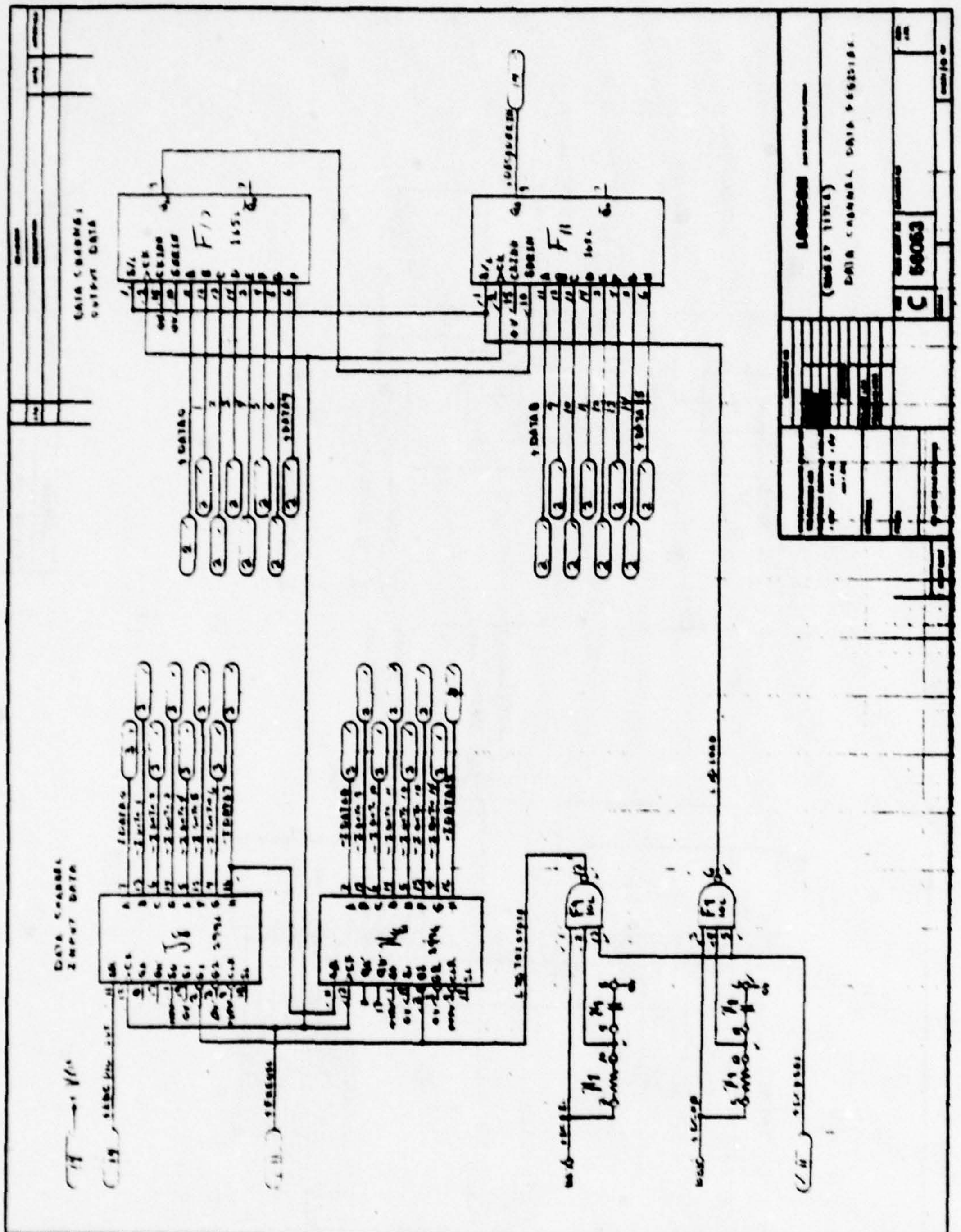


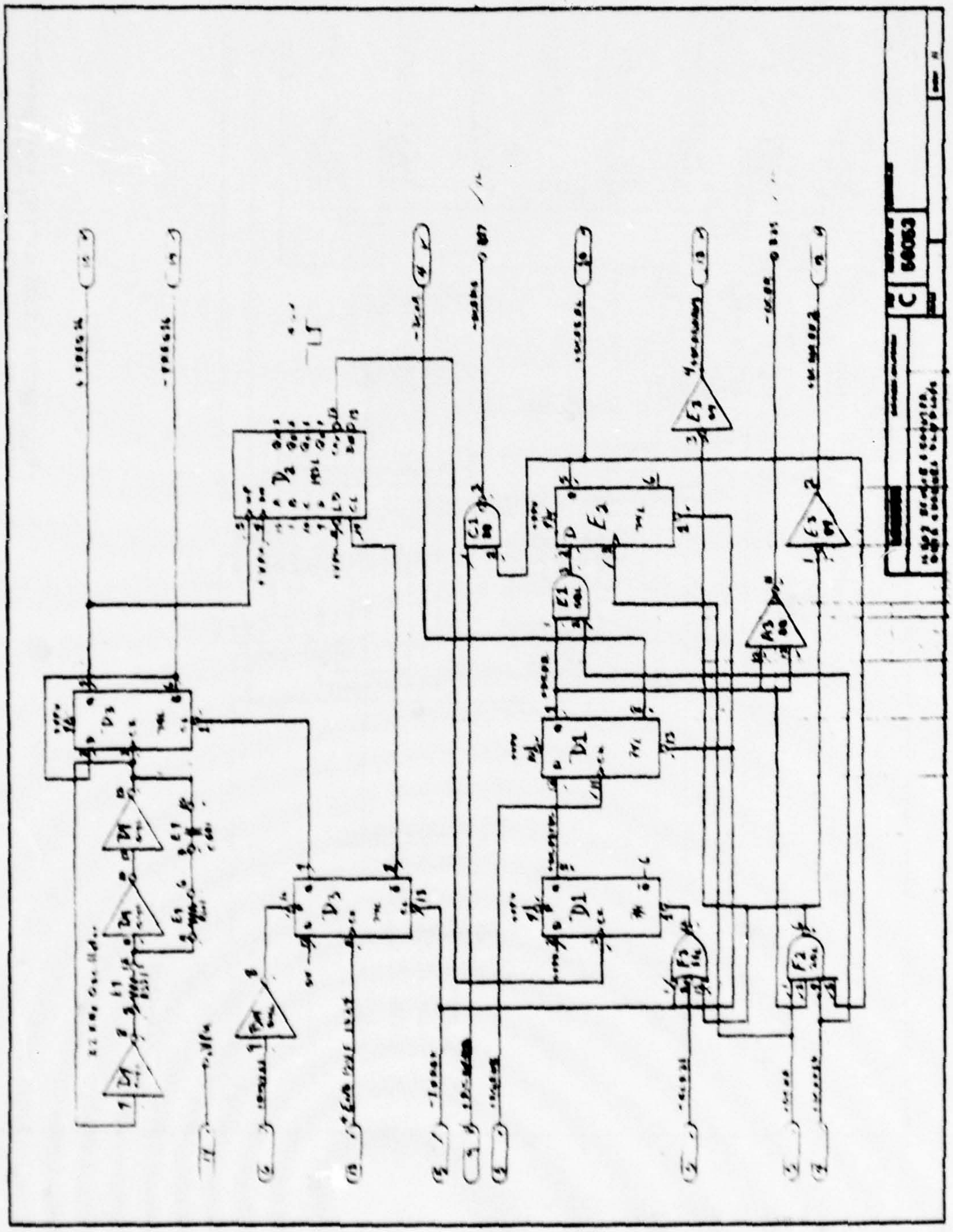


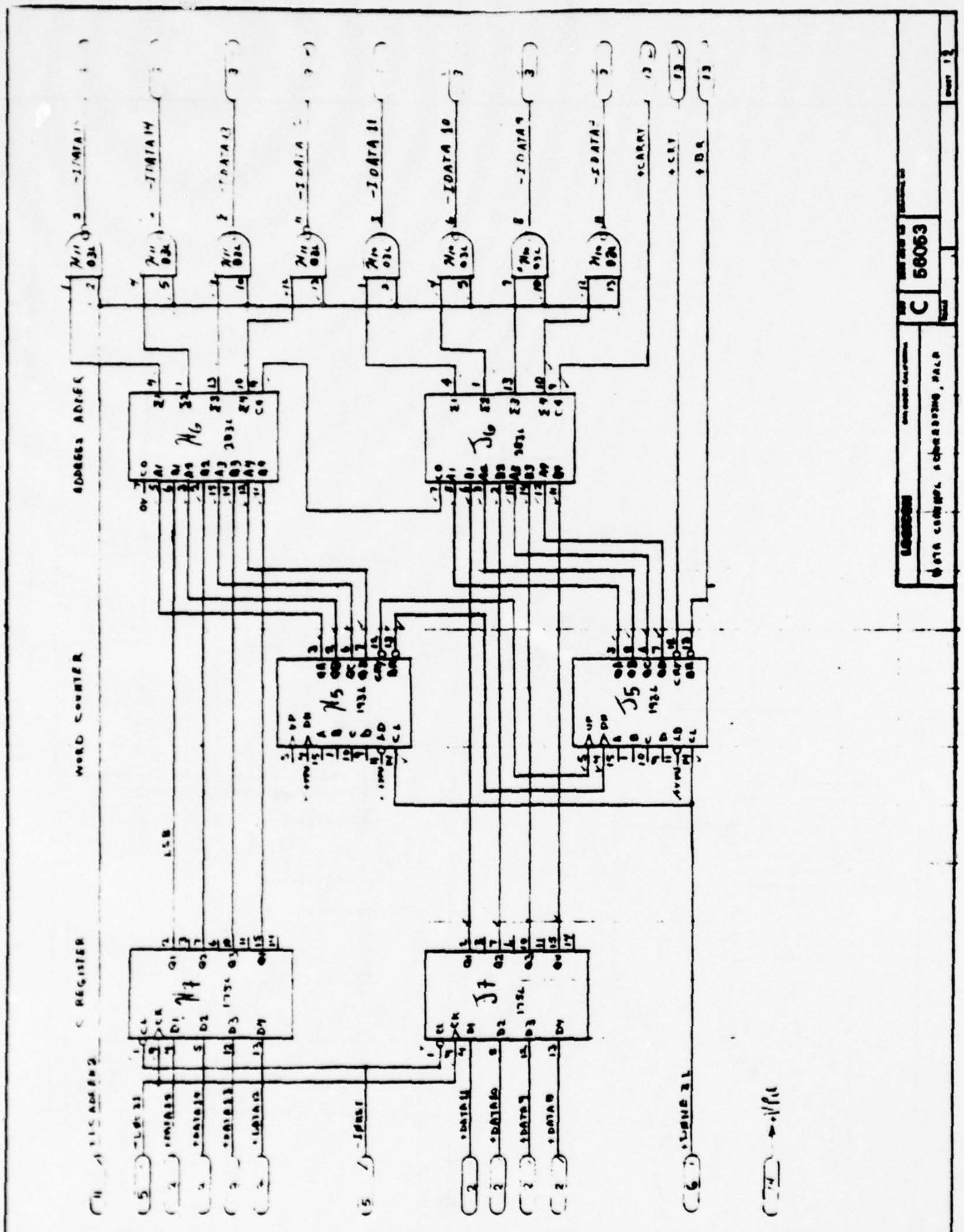

C8
D8
E8
F8
E7
E6

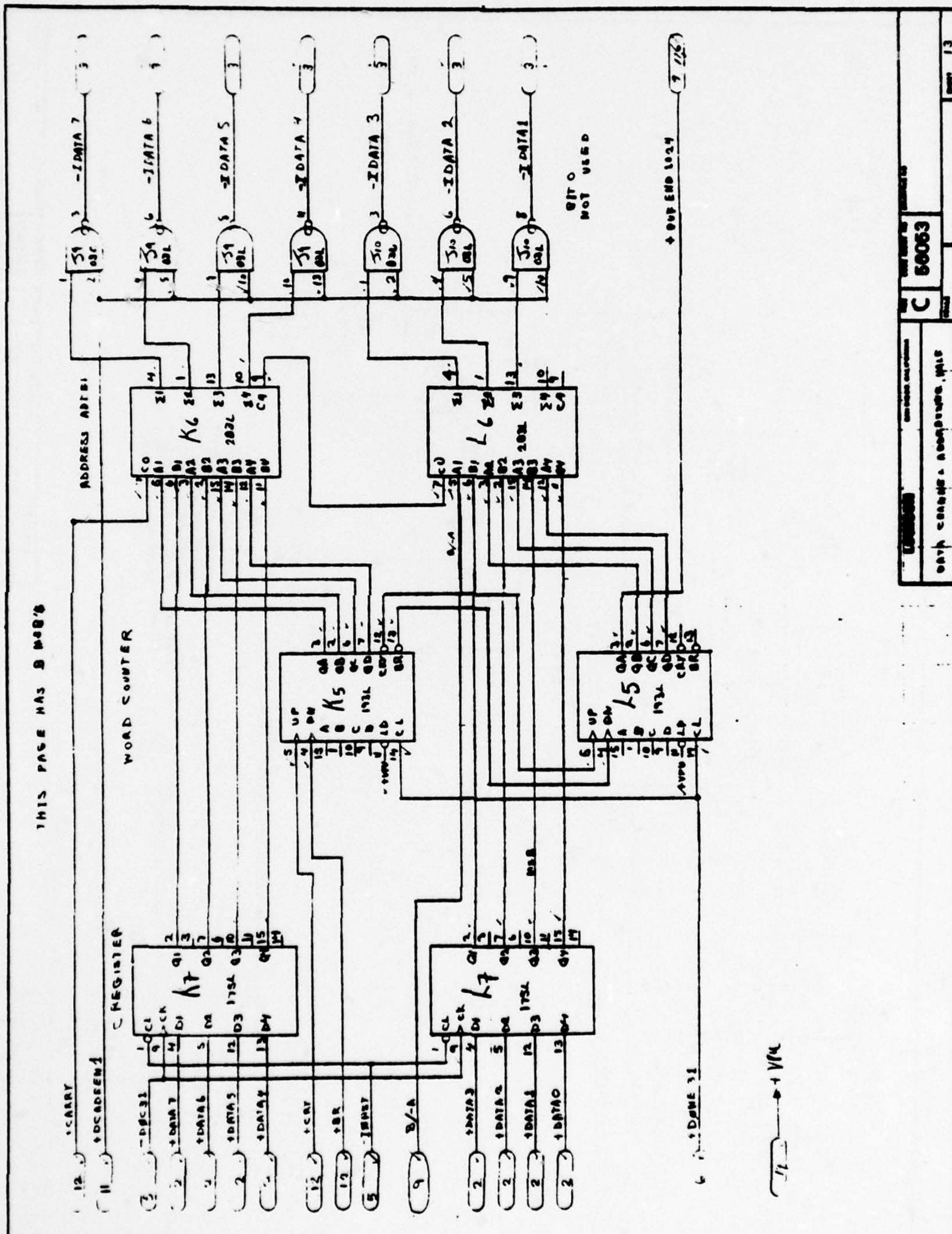


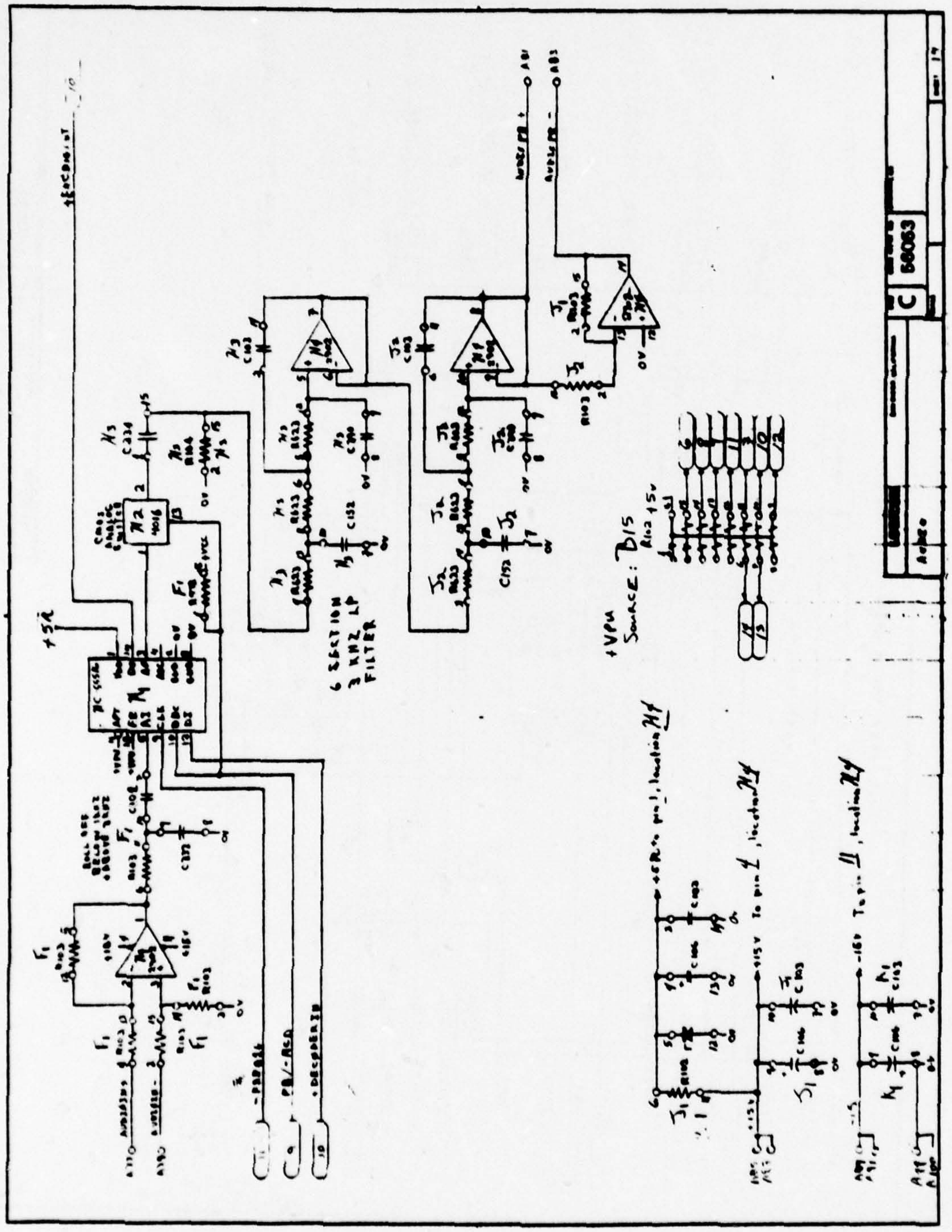






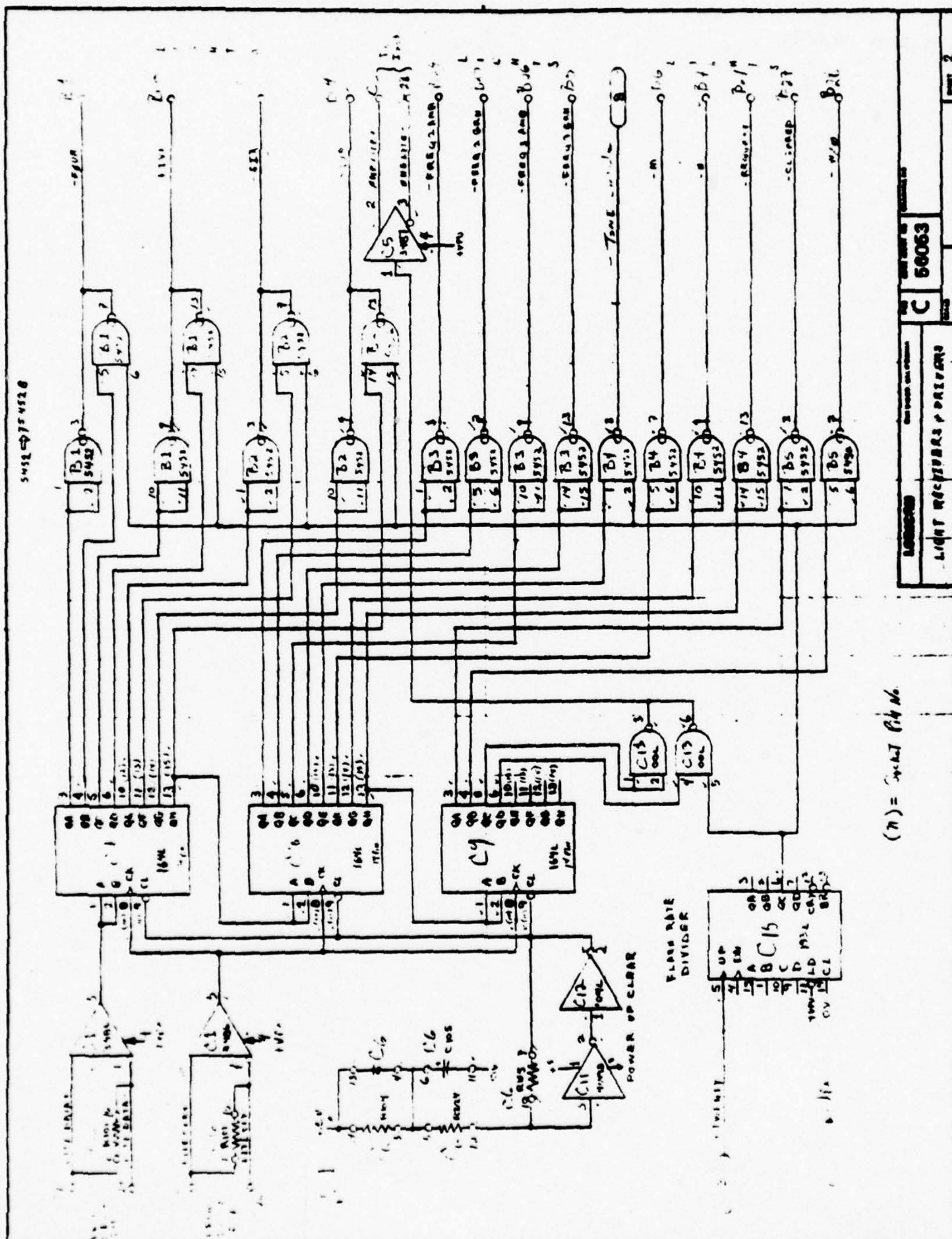

ADDRESS ADDER
WORD COUNTER
C REGISTER
C
58063

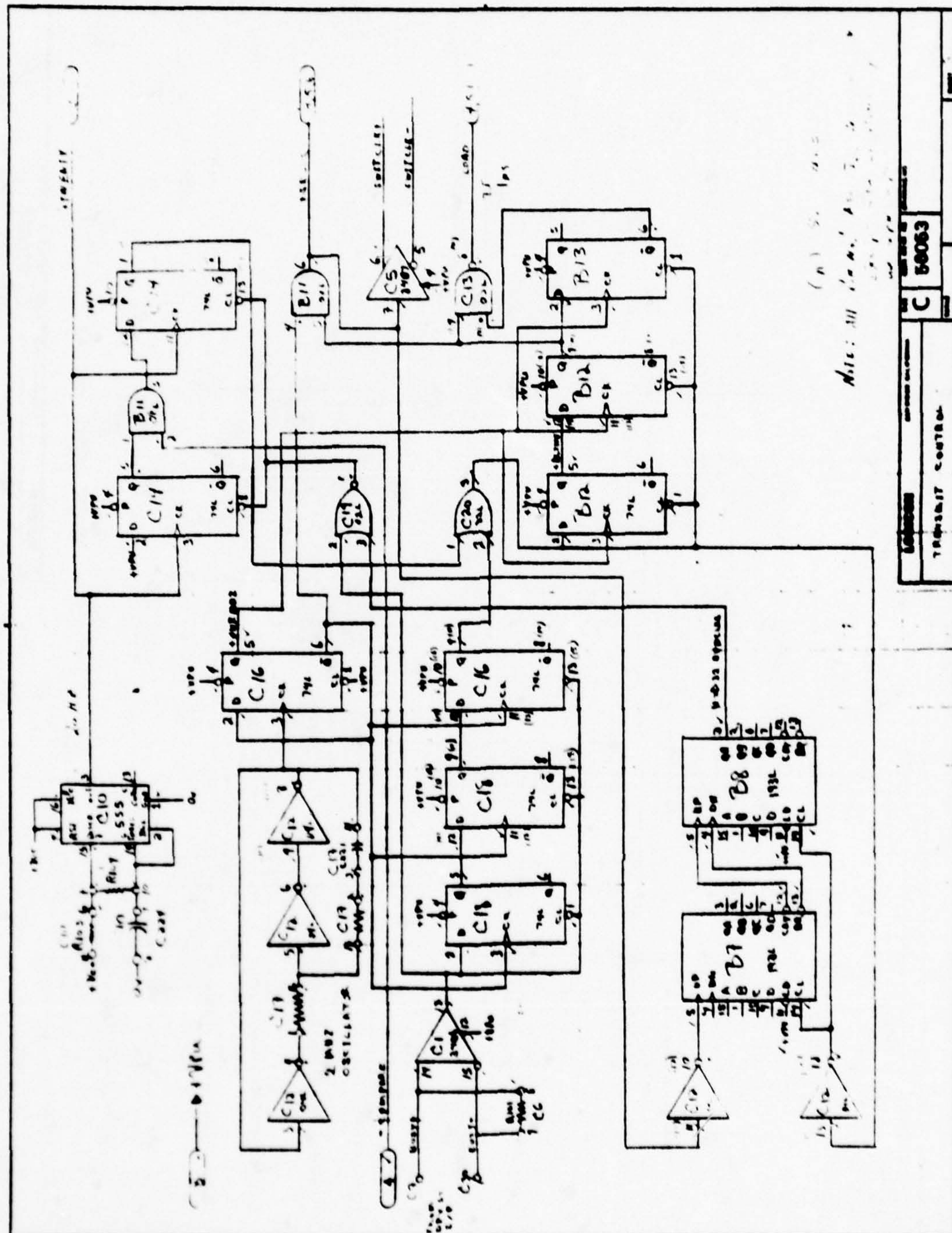


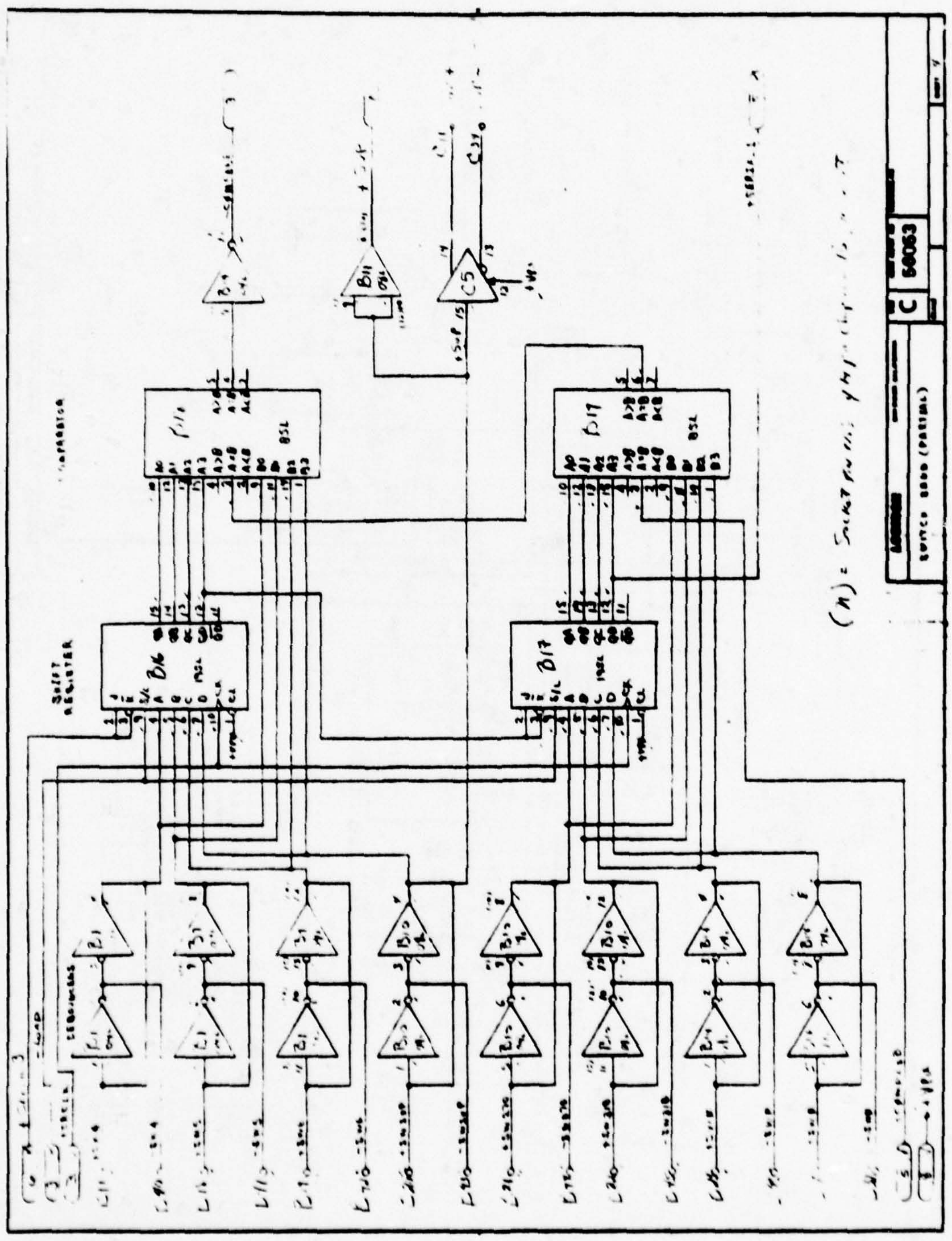


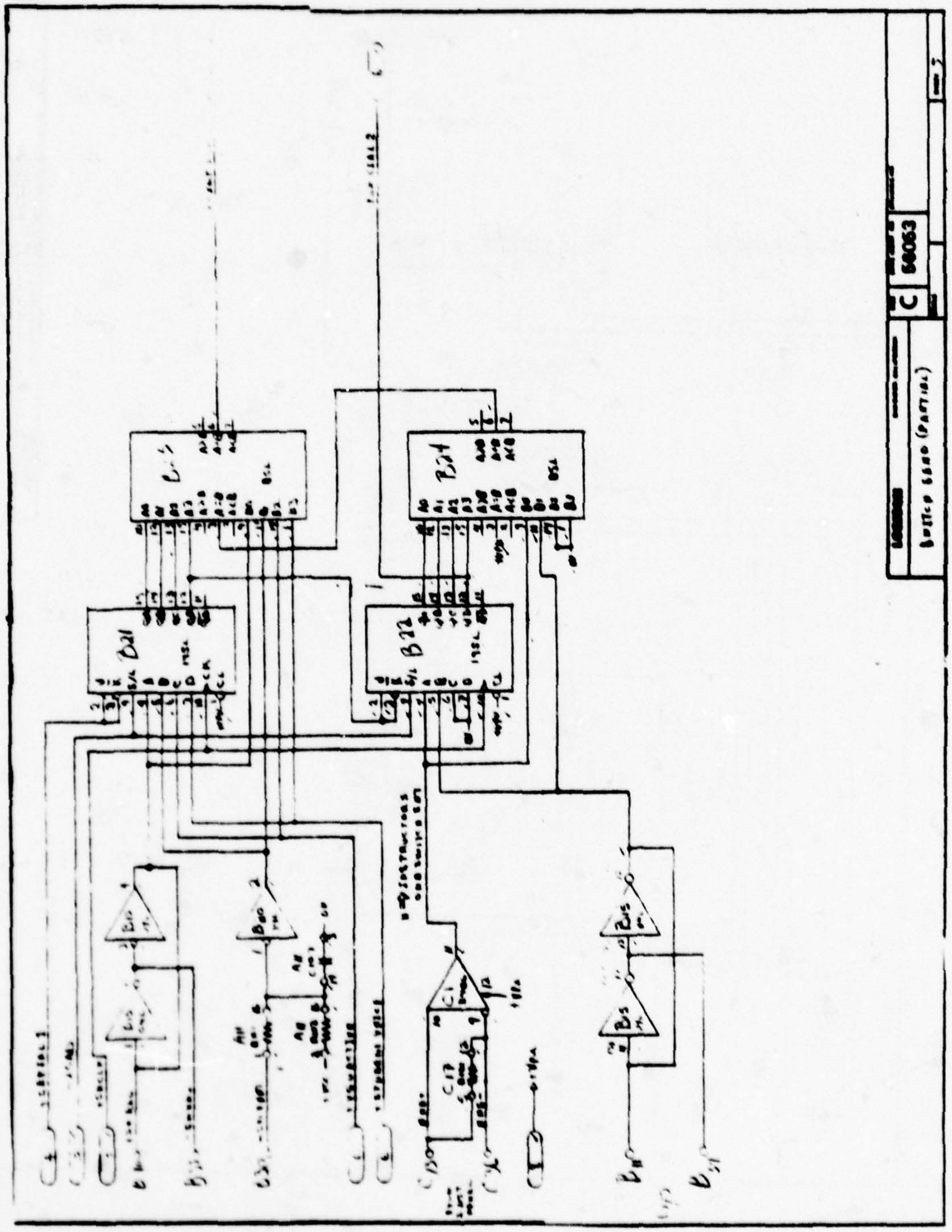
APPENDIX C

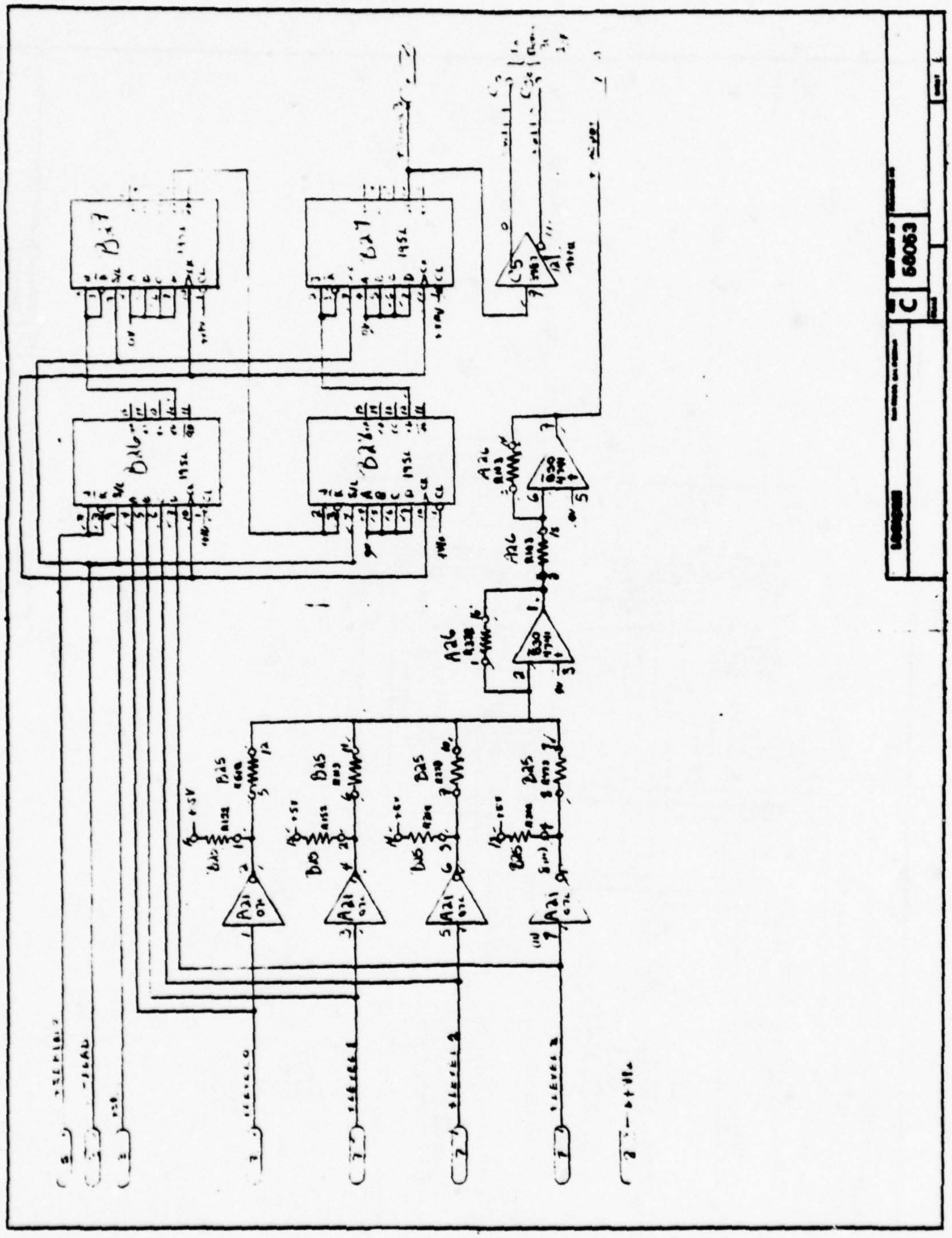
The working drawings in this appendix show the hardware design of the trainee panel.

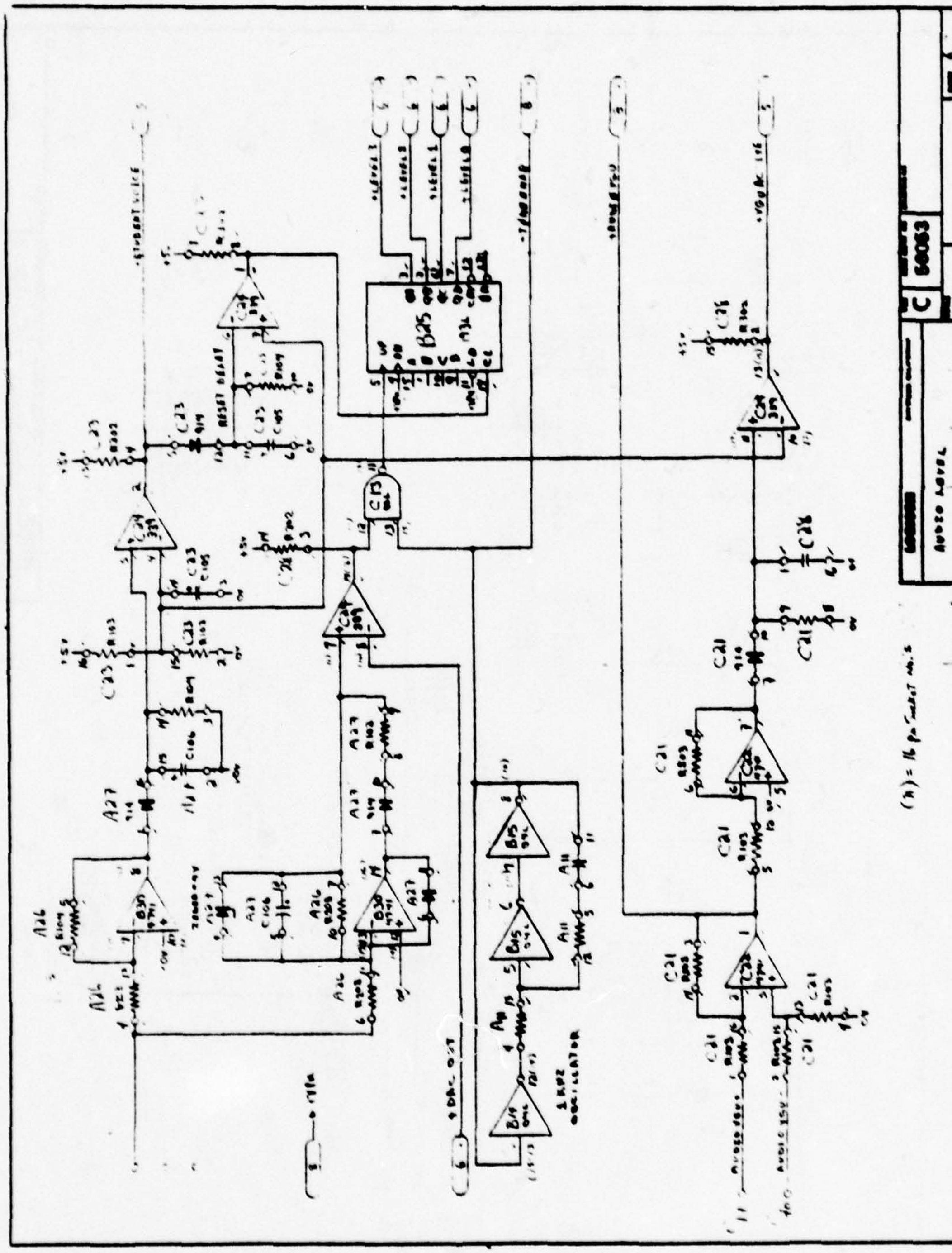


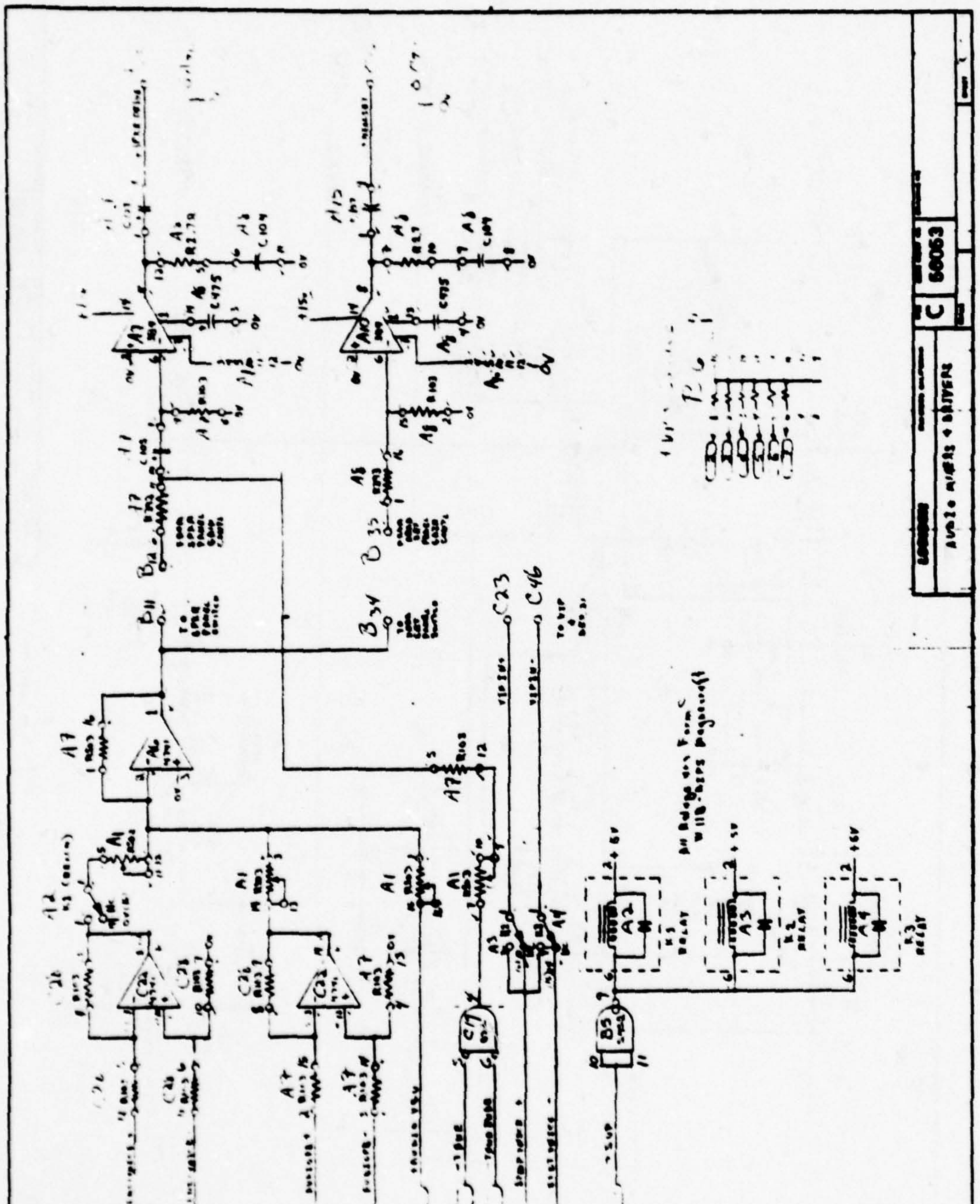

TRANSMIT CONTROL
C
58063





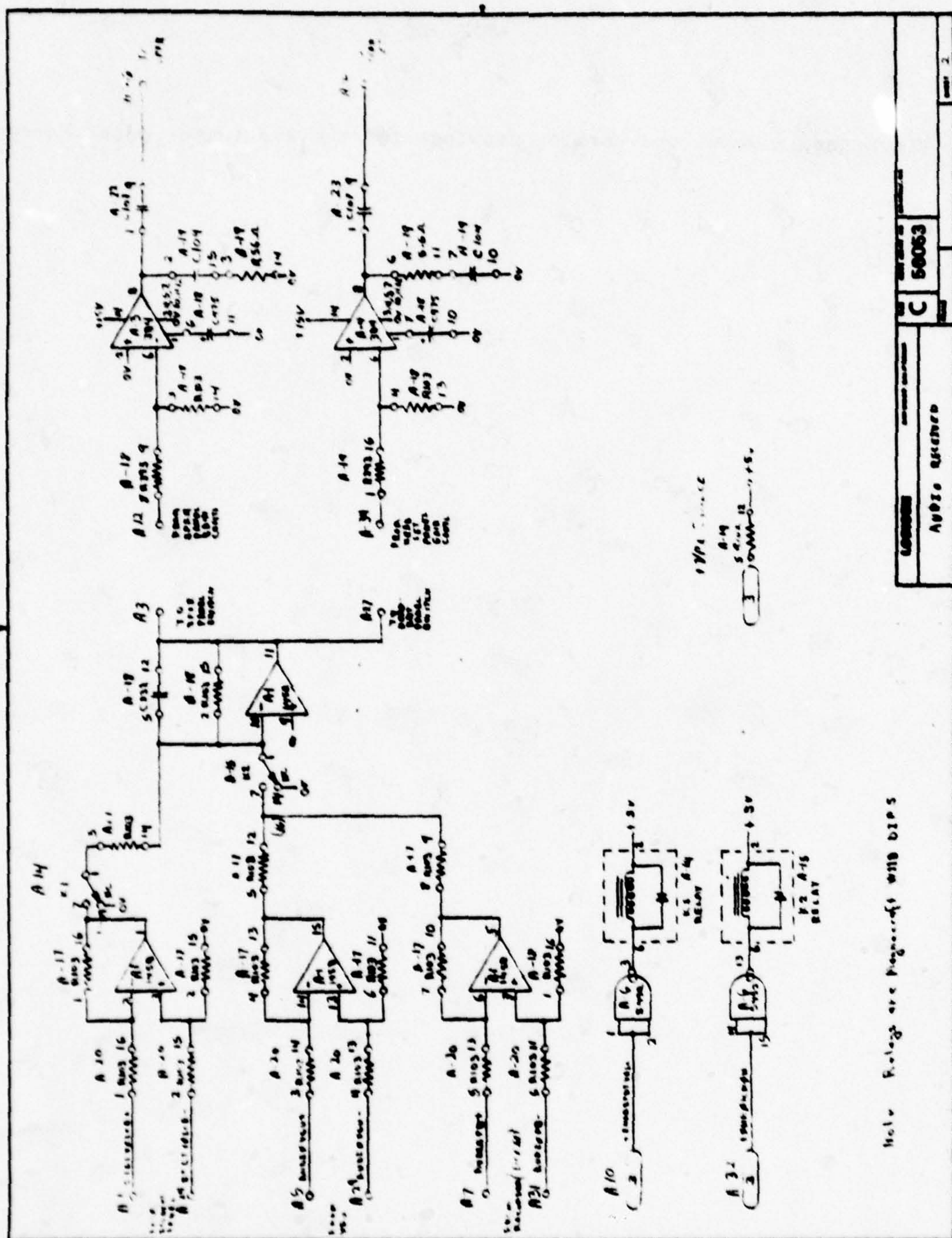


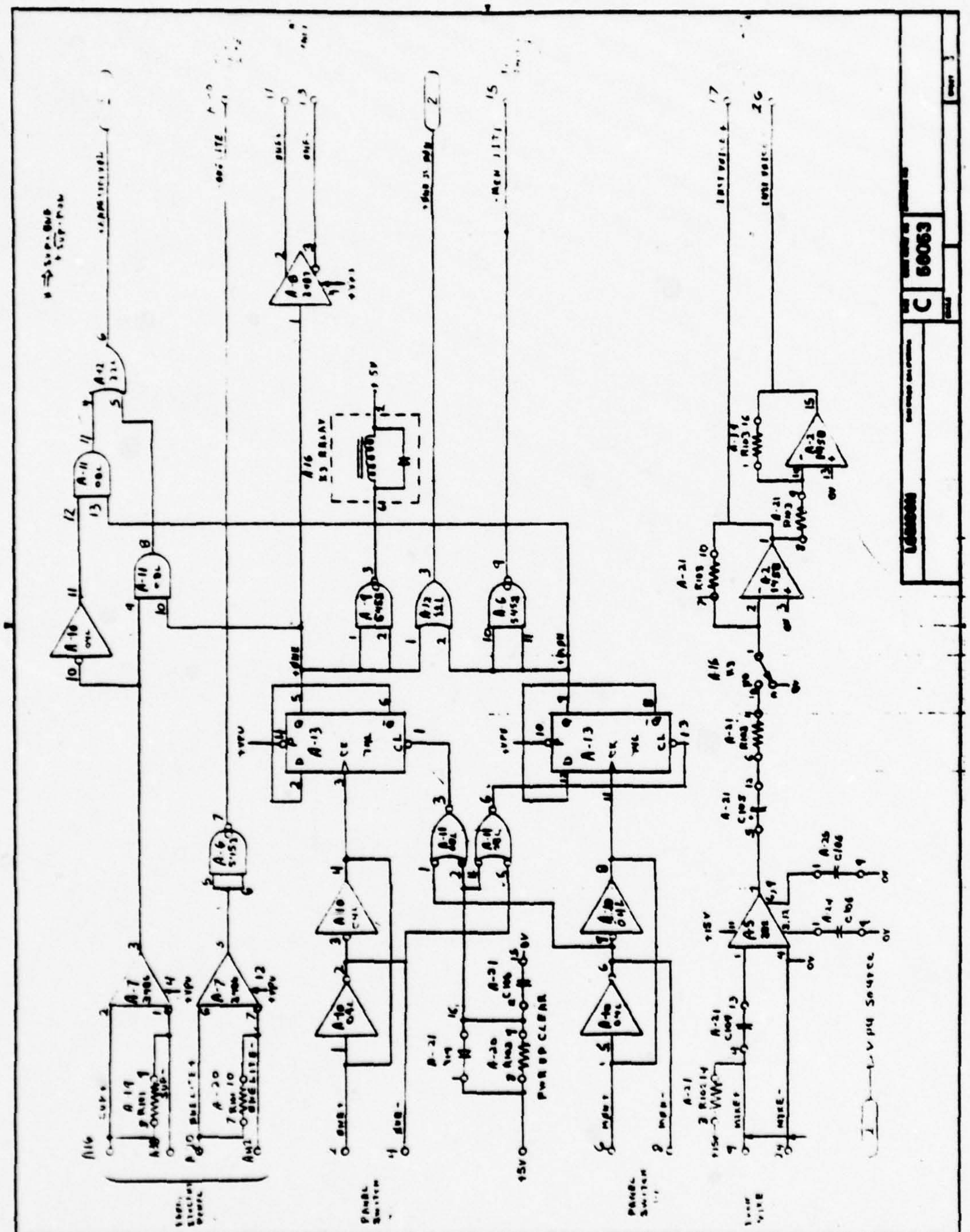




APPENDIX D

This appendix shows the working drawings for the instructor panel hardware.





Appendix E

Common Variable Definitions

Much of the communication between GCA-CTS routines will take place through labeled common. The variable names within a common block all begin with the same two letters. Table E-1 defines these naming conventions. The common blocks themselves are defined in tables E-2 through E-38.

Table E-1. Common Variable Naming Conventions

Variable Name Begins With	Common Block Names	Table Number
AC	ACFIX	E-2
BX	MAIL1, MAIL2	E-3,E-4
CL	CLOCK	E-5,E-6
CT	CTRLR	E-7
EM	EMERGE	E-8
EN	ENVIRON	E-9
EV	EVNTS, EVNT2	E-10,E-11
FN	FIL1, FIL2	E-12,E-13
FZ	FZ1	E-14
GZ	GZEC	E-15
KY	KEYS	E-16
MN	MENU1, MENU2	E-17,E-18
NC	DEV1, DEV2	E-19,E-20
PC	PCP	E-21
PF	PFSCR	E-22
PR	PRMPT	E-23
PT	PLT	E-24
PV	PMVC	E-25
P3	PZ3CM	E-26
RC	RECKON	E-27
RD	RDR	E-28
RP	PLAY	E-29
RQ	RQUE	E-30
SD	SPDGT	E-31
SK	SKED	E-32
SP	SPCH	E-33
SS	SHUSH	E-34
SV	SRV	E-35
TZ	TZC	E-36
VC	VOCIN	E-37
VL	VLID	E-38

Table E-2. ACFIX, CPU1 Aircraft Position Information

Mnemonic	Type	Content
ACALT	real	Altitude (feet)
ACRNG	real	Range (radar miles)
ACOFF	real	Distance from centerline (feet: if less than 0, left)
ACHDG	real	Heading (magnetic)
ACTYP	integer	Aircraft type
ACGYRO	logical	True: gyro working; false: gyro failure
ACSPD	real	Aircraft speed
ACGSPD	real	Aircraft ground speed
ACICE	logical	Control surface icing
ACHYDR	logical	Hydraulic failure
ACENG	logical	Single engine failure
ACAZN	integer	Azimuth zone
ACEZN	integer	Elevation zone
ACCS	integer	A/C call sign
ACYDOT	real	Rate of descent (ft/sec)
ACYDD	real	Vertical acceleraton (ft/sec^2)
ACXDOT	real	Lateral velocity (ft/sec)
ACXDD	real	Lateral acceleration (ft/sec^2)
ACNEW	logical	Start new approach flag

Table E-3. MAIL1, CPU1 XMT/REC Mailboxes

Mnemonic	Type	Content
BXSRMN	integer	User clock to SRMON IXMT key
BXIPB	integer	IPB to phase executives XMT key
BXRPL	integer	User clock to RDRPLY IXMT key
BXSPH	integer	SPOUT XMT key
BXACT	integer	User clock to ATRPLY IXMT key
BXPV	integer	Special timers, etc. to performance routines
BXRAT	integer	ACTOUT to RDACT mail
BXRC	integer	RPB interrupt service to SPDMP IXMT key
BXSKY	integer	RPB interrupt service to SPOUT IXMT key

Table E-4. MAIL2, CPU2 REC/XMT Mailboxes

Mnemonic	Type	Use
BXREC	integer	Signals voice input reception or timeout message
BXCOG	integer	Signals voice recognition made or timeout message

Table E-5. CLOCK1, CPU1 Clocks and Timers

Mnemonic	Type	Content
CL50	integer	50 minute class period timer
CLTGH	integer	Time to go home
CLTICK	integer	Half-second user clock
CL100	integer	100 msec user clock
CLTGT50	integer	Time 50 percent of target appears

Table E-6. CLOCK2, CPU2 Clock Common

Mnemonic	Type	Content
CLTICK	integer	Time in half second ticks from the start of the problem
CL100	integer	Time in 100 msec ticks from above time

Table E-7. CTRLR, CPU1 Model Controller Information

Mnemonic	Type	Content
CTCLR	integer	Clearance type
CTHOF	logical	True if feeder is to give handoff
CTGPP	integer	Correct guidepath position message
CTGPT	integer	Correct guidepath trend message, or -1
CTCRP	integer	Correct course position message
CTCRT	integer	Correct course trend message, or -1
CTRNG	integer	Correct range, including DH and OLT, or -1
CTOTHR	integer	Other final controller messages
CTEMERG	integer	Emergency waveoffs, etc., or -1
CTREL	logical	True if feeder is to fail to relinquish radio frequency until requested to do so
CTNGR	logical	True if feeder is to give no gyro handoff
CTFREQ	integer	Frequency for this problem 1:270.8; 2:318.8
CTSPH	integer	Last feeder controller output
CTSPT	integer	Time CTSPH is over
CTON	logical	True if controller is giving the demo
CTPHZ	integer	Phase of flight (1, 2 or 3)
CTDEV	integer	Requested final controller speech device (1:$VRO, 2:CRT, 3:AUDIO)
CTACK	logical	True if pilot is still acknowledging controller messages
CTGPOS	logical	True if position message on glidepath is legal
CTCPOS	logical	True if position message on course is legal
CTDOWN	logical	True if "Begin descent" has been given
CTATHT	logical	True if after decision height

Table E-8. EMERGE, CPU1 Emergency Data

Mnemonic	Type	Content
EMGYFL	logical	True if gyro failure is to occur
EMGYR	real	Range at which gyro failure occurs
EMICFL	logical	True if icing is to occur
EMICR	real	Range at which icing is to occur

Table E-9. ENVIRON, CPU1 APE Environment

Mnemonic	Type	Content
ENWSP	real	Wind speed
ENWHDG	real	Wind direction
ENFLUCT	real	Percent wind fluctuation
ENWZ	real	Present wind longitudinal component
ENWX	real	Present wind lateral component
ENCNDN	integer	Environmental condition number
ENSEED	integer	Pseudo-random number seed for use by environmental simulation

Table E-10. EVNTS, CPU1 Events (All Are Parameters)

Mnemonic	Value	Use
EVPHZ	1	Wakeup phase executive (1, 2, or 3)
EVVIN	2	Signal end of student voice input
EVVRO	3	Signal end of Votrax output
EVPNL	4	Signal input from student panel
EVKEY	5	Signal keyboard input (student)
EVZEC	6	Wakeup training executive
EVSPT	7	SPOUT finished

Table E-11. EVNT2, CPU2 Events

Mnemonic	Value	Use
EVPHZ	1	Wakeup phase executive
EVVIN	2	Signal end of student voice input
EVKEY	5	Signal student keyboard input
EVVRPD	3	Signals end of VRP load

Table E-12. FIL1, CPU1 File Names

Mnemonic	Dimension	Content
FNSYL	7	Syllabus file name
FNDR	7	Student directory file name
FNTXT	4	Text file name (to CPU1)
FNPHZ	4	Phase problem file names

Table E-13. FIL2, CPU2 File Names

Mnemonic	Dimension	Content
FNIFP	4	IFP collection file name
FNVRP	4	VRP collection file name
FNSPK	4	Trainee independent VDC/SUS data file name

Table E-14. FZ1, CPU1 Phase 1 Information

Mnemonic	Type	Content
FZPTR	integer array	Phase 1 file position pointer
FZINP	integer	Array (1:64) "A" input format, content of current phase 1 record
FZSKP	integer	Number of records to be skipped upon timeout of wait condition
FZFRZ	integer	Freeze key message number
FZSUB	integer array	Stores five abnormal return skip record values. These skips are in reference to the normal return point. Array (1:5, 1:5) allows five levels of nesting.
FZRET	integer array	Store present value of the file position pointer. Array (1:5) allows five levels of nesting.
FZFLG	integer array	Holds flag values. Array (1:10) provides 10 flags to be manipulated by the phase 1 task file.
FZSRV	logical	If true servo has been activated, if false servo is frozen

Table E-15. GZEC, CPU1 GCA-CTS Executive

Mnemonic	Type	Content
GZNR	integer	Number of repeats of this problem
GZFRZ	logical	Error termination, phase 2
GZTRY	integer	Number of tries, this problem, phase 2
GZPHZ	integer	Phase
GZPAS	logical	Used by SCORE to inform phase 3 that criteria for advancement have been met
GZSOFL	logical	Where true, IPBIN1 sends signoff request to TZEC.
GZCHAL	logical	Challenge phase 3 (skipping phase 2)
GZSEED	integer	Seed for next pseudo-random number
GZRPL	integer	Set to type of replay if mandatory replay request was made
GZPILL	integer array	Remedial training exercises proposed by SELECT
GZPRUN	logical	True when P run is in progress

Table E-16. KEYS, CPU1 Indicators of Buttons Depressed at Trainee and Instructor Stations

Mnemonic	Type	Description
KYHOLD	logical	When true, interrupt service does not respond to buttons
KYDIA	integer	DIA word from student panel
KYDIB	integer	DIB word
KYIC3	logical	True while ICS button 3 is selected
KYIC5	logical	True while ICS button 5 is selected
KYIC7	logical	True while ICS button 7 is selected
KYICS	logical	True while ICS button SUPER is selected
KY27F	logical	True while radio frequency 270.8 is selected
KY31F	logical	True while radio frequency 318.8 is selected
KY27M	logical	True while radio monitor 270.8 is selected
KY31M	logical	True while radio monitor 318.8 is selected
KYREQ	logical	True when clearance is requested
KYMIKE	logical	True while mike is keyed
KYLVL	integer	Level of speech input
KYCLR	logical	True while clearance light is on
KYWOL	logical	True while waveoff light is on
KYWOB	logical	True when waveoff button is depressed
KYSPH	logical	True while student is speaking

Table E-17. MENU1, CPU1 Menus

Mnemonic	Type	Content
MNST	integer array	Student menu
MNIN	integer array	Instructor menu

Table E-18. MENU2, CPU2 Menus

Mnemonic	Type	Content
MNST	integer array	Student menu
MNIN	integer array	Instructor menu

Table E-19. DEV1, CPU1 Channel Numbers

Mnemonic	Channel	Use	Open
NCSYL		Syllabus file	always
NCSR		Student records	always
NCPHZ		Problem files, all phases also remedial training file	during phase
NCSCR		Scratch file	always
NCLPT	12	Printer	always
NCVRO	5	Votrax	always
NCFRZ	13	FRAZ file	always
NCOL		Overlay file	always
NCAO	10	Instructor CRT	always
NCAIN	11	Instructor keyboard	always
NCRPLY		Replay file, radar data	phase 3, P-run
NCRPAT		Replay file, activity data	phase 3, P-run
NCDV		Digitized voice file (student)	phase 3, P-run phase 1
NCCDV		Canned digitized voice file	during phase
NCBUG		Bug file	always
NCDP1		IPB input	always
NCDPO		IPB output	always

Table E-20. DEV2, CPU2 Channel Numbers

Mnemonic	Channel	Use	Open
NCTXT		Text files, all phases	during phase
NCVF		Voice data file	during phase
NCTAR		Voice data training arrays	phase 1
NCAO	10	Student CRT	always
NCAIN	11	Student keyboard	always
NCDPI		IPB input	always
NCDPO		IPB output	always
NCBUG		Bug file	always

Table E-21. PCP, CPU2 Aircraft Update Information

Mnemonic	Type	Description
PCAMVD	logical	True if azimuth target moved
PCEMVD	logical	True if elevation target moved

Table E-22. PFSCR, CPU1 Current PMV Switches

Mnemonic	Type	Content
PFS01	logical	True if phase 2 to freeze on error of PV01, or if phase 3 to score PV01 first.
. . .		. . .
PFQUE	integer array	Holds the queue of controller messages to be started by PUT
PFHDG	integer	Last assigned heading
PFGPP	integer	Last glidepath position given
PFGPT	integer	Last glidepath trend given

Table E-23. PRMPT, CPU1 Prompt Information

Mnemonic	Type	Content
PRATL	integer array	Array (1:N) in phrase identification order. Words are set to 1 to indicate the phrase has been authorized for prompting by model controller.
PRDEV	integer	Last prompt device for final controller 1 = $VRO 2 = CRT 3 = AUDIO
PRDNE	logical	True, if nothing is being said by the Votrax or digitized audio
PRQUE	integer array	Array (1:30). Queue of messages to be said. First word is output device (negative value). Following words are phrase numbers.
PRENT	integer	PRQUE entry position pointer
PRSAY	integer	PRQUE start of message to be output pointer
PRES	integer	Last message to be said or being said
PRAC	logical	True, if array being accessed and pointers changed

Table E-24. PLT, CPU1 Pilot Information

Mnemonic	Type	Content
PTSEED	integer	Seed for pseudo-random number generator
PTYP	integer	Pilot type (1:best, 5:worst)
PTFLT	integer	Type of flight (1:normal, 2:azimuth flight restricted, 3:elevation flight restricted)
PTLZN	integer	If PTFLT = 2, left azimuth zone If PTFLT - 3, lower elevation zone
PTUZN	integer	If PTFLT = 2, right azimuth zone If PTFLT - 3, upper elevation zone
PTLOW	real	If PTFLT = 1, range at which aircraft descends to low altitude alert level, or 0.
PTAPR	integer	Type of approach
PTRNG	real	Range at which approach is to terminate
PTRESP	integer	Last pilot response
PTBYE	integer	Time when pilot will execute lost communications procedure
PTYDOT	real	Pilot's estimate of rate of descent error (ft/sec)
PTYERR	real	Pilot's estimate of altitude error (ft)
PTXDOT	real	Pilot's estimate of lateral velocity error(ft/sec)
PTXERR	real	Pilot's estimate of lateral offset (ft)
PTCORS	integer	Latest course advisory word
PTPATH	integer	Latest glidepath advisory word
PTMISS	logical	Missed approach flag
PTWEEL	logical	Wheels down flag
PTDES	logical	Begin descent flag
PTRANG	integer	Latest range advisory, miles

Table E-25. PMVC, CPU1 PMV Scores

Mnemonic	Type	Content
PVN**	integer array	Allowable error scores
PVE**	integer array	Observed error scores
PV01	integer	Handoff composite
PV02	integer	Radio check composite
PV03	integer	Turn to final composite
PV04	integer	Approaching glidepath composite
PV05	integer array	Heading advisories composite
PV06	integer array	Azimuth position and trend composite
PV07	integer array	Glidepath position and trend composite
PV08	integer	Range call composite
PV09	linteger	Decision height composite
PV10	integer	Clearance composite
PV11	integer	Landing threshold composite
PV12	integer	Handoff, rollout composite
PV13	integer	No gyro composite
PV14	integer array	No gyro heading corrections
PV15	integer	Emergency waveoffs
PV16	integer	Low altitude alert
PV17	integer	Transmission break

Table E-25. PMVC, CPU1 PMV Scores (Cont)

Mnemonic	Type	Content
PV18	integer array	Transmission rate composite
PV19	integer	Radar alignment composite Note: PV20-23 are not scored
PV20	integer array	Ratio glidepath to course messages
PV21	integer	Number of times aircraft crosses centerline
PV22	integer array	Used for average transmission rate
PV23	integer	Number of times course headings were not properly stylized

Table E-26. PZ3CM, CPU1 PZ3B Local Common

Mnemonic	Type	Content
P3S(1)	integer array	% Starting position 1:minimum fuel
P3S(2)		% Starting position 2:right base (cumulative)
P3S(3)		% Starting position 3:straight in (cumulative)
P3S(4)		% Starting position 4:left base (cumulative)
P3SV(1)	integer array	% Heading variation on position 1 starts
P3SV(2)		% Heading variation on position 2 starts (cumulative)
P3SV(3)		% Heading variation on position 3 starts (cumulative)
P3SV(4)		% Heading variation on position 4 starts (cumulative)
P3SP(1)	integer array	% Slow aircraft
P3SP(2)		% Medium aircraft (cumulative)
P3SP(3)		% Fast aircraft (cumulative)
P3P(1)	integer array	% Type 1 pilot (best)
P3P(2)		% Type 2 pilot (cumulative)
P3P(3)		% Type 3 pilot (cumulative)
P3P(4)		% Type 4 pilot (cumulative)
P3P(5)		% Type 5 pilot (cumulative)
P3A(1)	integer array	% Full stop
P3A(2)		% Low approach (cumulative)
P3A(3)		% Touch & go (cumulative)

Table E-26. PZ3CM, CPU1 PZ3B Local Common (Cont)

Mnemonic	Type	Content
P3NGY(1)	integer array	Percent no gyro approaches
P3NGY(2)		Cumulative percent ordinary approaches
P3CL(1)	integer array	Percent clearance given at first request
P3CL(2)		Cumulative percent continue then clear at 2 miles
P3CL(3)		Cumulative percent no response
P3CL(4)		Cumulative percent waveoff
P3CL(5)		Cumulative percent clearance given then cancelled
P3WN(1)	Integer array	Percent light and variable winds
P3WN(2)		Cumulative percent 190° at 10 kts
P3WN(3)		Cumulative percent 190° at 20 kts
P3WN(4)		Cumulative percent 250° at 10 kts
P3WN(5)		Cumulative percent 250° at 20 kts
P3LA(1)	integer array	Percent low altitude alert conditions met
P3LA(2)		Cumulative percent low altitude alert conditions not met
P3MS(1)	integer array	Percent minimum separation violation runs
P3MS(2)		Cumulative percent non minimum separation violations
P3ICE(1)	integer array	Percent runs with icing
P3ICE(2)		Cumulative percent no icing

Table E-26. PZ3CM, CPU1 PZ3B Local Common (Cont)

Mnemonic	Type	Content
P3HYF(1)	integer array	Percent runs with hydraulic failure
P3HYF(2)		Cumulative percent no hydraulic failure
P3ENG(1)	integer array	Percent runs with single engine failure
P3ENG(2)		Cumulative percent no engine failure

Table E-27. RECKON, CPU2 Voice Recognition Common Block

Mnemonic	Type	Content
RCGPP	integer	Correct glidepath position mask, or -1
RCGPT	integer	Correct glidepath trend mask, or -1
RCCRP	integer	Correct course position mask, or -1
RCCRT	integer	Correct course trend mask, or -1
RCRNG	integer	Correct range, including DH and OLT, or -1
RCEMERG	integer	Emergency related message mask, or -1
RCOTHR	integer array	Other final controller masks, or -1 (All of above data initialized to -1)
RCRES	integer array	Array (1:6). Resolution masks for the above controller messages
RCPHS	integer array	Array (1:6). Phase of flight masks to be used as resolution masks. These mask values are data initialized
RCBF	integer array	Recognition information buffer: Word 1 = 11, identifies this as a recognition block
Word 2		Time of LP_4 in .5 second ticks from the start of the problem
Word 3		Time in 100 msec. ticks from above timer
Word 4		First choice message recognized
Word 5		Heading flag, or -1
Word 6		Wind flag, or -1
Word 7		Second recognition choice, or -1
Word 8		Missed approach flag, or -1
RCFZIS	integer	Phase of flight 1 = initial handoff 2 = body of approach 3 = final handoff

Table E-28. RDR, CPU1 Radar Information

Mnemonic	Type	Content
RDAZR	logical	Azimuth radar display (on if true)
RDAZS	logical	Azimuth servo (on if true)
RDELR	logical	Elevation radar display
RDELS	logical	Elevation servo
RDSVAZ	integer	Servo position, azimuth antenna
RDSVEL	integer	Servo position, elevation antenna
RDCLR	integer	Centerline alignment zone
RDTDR	integer	Touchdown alignment zone
RDRNG	integer	Range alignment zone (concerns touchdown)
RDALT	integer	Altitude (Y_E) of target — screen coordinates
RDCRS	integer	Course (Y_A) of target — screen coordinates
RDRAN	integer	Range (X) of target — screen coordinates
RDMIL	integer	Last range info from APE in miles

Table E-29. PLAY, CPU1 Replay Common

Mnemonic	Type	Content
RPTIME	integer	Time in .5 second ticks from run initiation
RPTSP	integer	Time speech output to stop, .5 sec ticks
RPTDLY	integer	Offset at which speech output to stop
RPRBF	integer	Current radar buffer
RPPTR	integer	Pointer into radar information buffer
RPSP1	logical	If true, buffer 1 is full of speech data
RPSP2	logical	If true, buffer 2 is full of speech data
RPSBF	integer	Current speech buffer
RP100	integer	100 msec clock for timing within a .5 second tick
RPABF	integer	Current activity information buffer
RPATR	integer	Pointer into activity buffer
RPSP3	logical	If true, speech holding buffer is full of speech data
RPINIT	integer	1 at initialization or restart of replay, else 2
RPALK1	integer	Activity buffer RPLACT lock, buffer A
RPALK2	integer	Activity buffer PMS lock, buffer A
RPBLK1	integer	Activity buffer RPLACT lock, buffer B
RPBLK2	integer	Activity buffer PMS lock, buffer B
RPTAT	integer	Time next activity occurs

Table E-30. RQUE, CPU1 Queue of Tasks to be Scheduled at Given Ranges

Mnemonic	Type	Content
RQTSK	integer array	Array of task LOCO entry points
RQRNG	real array	Array of ranges for task scheduling
RQLNK	integer array	Linked list used to order ranges
RQPTR	integer	Pointer to next task to be started

Table E-31. SPDGT, CPU1 Digitized Speech Buffers

Mnemonic	Type	Content
SDBF1	integer array	Digitized speech buffer 1
SDBF2	integer array	Digitized speech buffer 2

Table E-32. SKED, CPU1 Scheduling Information

Mnemonic	Type	Content
SKTEN	integer array	Linked list of tasks to be called with times
SKTNX	integer	Next task to be called in SKTEN
SKTAV	integer	Available slot pointer in SKTEN
SKREN	integer array	Linked list of tasks to be called, messages
SKRNG	integer array	Array of ranges at which tasks in SKREN are to be called
SKRNX	integer	Next task to be called
SKRAV	integer	Available slot pointer in SKRNG

Table E-33. SPCH, CPU2 Speech Associated Variables

Mnemonic	Type	Content
SPAT	integer array	Array (1:N), each word in the array is associated with a final controller phrase for the current student. Each word acts as a flag for SUS and audio prompts. If 0 = No VRP 1 = VRP formed Array is in phrase id order
SPVAL	integer array	Array (1:N) in phrase id order. Each entry yields the validated percentages for the corresponding phrase VRP. It is set to zero before validation occurs.
SPDEV	integer	Prompt device last used. 1 = $VRO 2 = CRT 3 = Audio
SPLOD	logical	True if VRPs have been loaded
SPFLG	integer	<0 = VDC, 0 = none, >0 = SUS on
SPLVL	integer	Voice input level
SPID	integer array	Array (1:N) is in phrase id order. Each entry is the phrase id
SPVRP	integer array	VRP file position pointer. Array (2,N) is in phrase id order. Each entry is a pointer to the start of the VRP
SPNUM	integer array	Array (1:N) is in phrase id order. SPNUM (1:N) indicates # of IFPs necessary to form a VRP
SPLST	integer array	Array (1:7). SPLST (1) may indicate either % validation or a phrase number. SPLST (2-7) are either phrase numbers or set to zero.
SPIFP	integer array	Array (2,N) is in phrase id order. File position pointers to IFP storage location.

Table E-34. SHUSH, Speech Understanding Related Variables in CPU1

Mnemonic	Type	Content
SSBFA(1)	integer array	Buffer lock Bit 13-15-0: Buffer being filled Bit 15-1: Buffer ready Bit 14-1: APE release Bit 13-1: PMS release
Word 2		Time of LP_4 in .5 second ticks from the start of the problem
Word 3		Time in 100 msec. ticks from above
Word 4		First choice message understood
Word 5		Heading, if any (turns, wind)
Word 6		Call sign indicator, or wind speed Bits 13-15: Call sign 1-4
Word 7		Second choice message understood, or -1
Word 8		Unused
SSBFB	integer array	As above
SSBFO(1)	integer array	Serves as input to activity replay file. Word 1 equals: 0, not in use 1, awaiting next phrase understood 3, ready to output
Word 2-7		As in SSBFA and SSBFB
Word 8	T,F	T if correction was applied to this phrase
SSCAT	integer array	Gives status for each message type task, Array (1:6)
Word 1		Heading status, or -1
Word 2		Wind status, or -1
Word 3		Missed approach status, or -1
Word 4		Digit status, or -1
Word 5		Misrecognitions status, or -1

Table E-34. SHUSH, Speech Understanding Related Variables in CPU1 (Cont)

Mnemonic	Type	Content
Word 6		Other status, or -1
SSBFW	integer array (6x8)	SUS working buffers. Same format as SSBFA and SSBFB for each of the above tasks
SSDIG	integer array	Array (1:3, 1:2). Each entry is a digit to be stored. The digit task is the only one which actually fills it.
SSAZN	integer	Aircraft azimuth zone at time of student input
SSHDG	integer	Aircraft heading at time of student input
SSRNG	integer	Aircraft range at time of student input
SSBFI	integer array	Array (1:8). SUS input buffer. Word 1 = 11, identifies this as a recognition block
Word 2		Time of LP_4 in .5 second ticks
Word 3		Time in 100 msec. ticks from above time
Word 4		First choice message
Word 5		Heading flag, or -1
Word 6		Wind flag, or -1
Word 7		Second recognition, or -1
Word 8		Missed approach flag, or -1

Table E-35. SRV, CPU2 Servo Information

Mnemonic	Type	Content
SVLOX	integer	Servo limit, lower X
SVLOY	integer	Servo limit, lower Y
SVHIX	integer	Servo limit, upper X
SVHIY	integer	Servo limit, upper Y
SVSETX	integer	Location of servo, X-plane
SVSETY	integer	Location of servo, Y-plane
SVZN(4)	integer	Displacement to add to reflector position
SVSTDA(2)	integer	Standard position azimuth touchdown reflector
SVSTDE(2)	integer	Standard position elevation touchdown reflector
SVSCL(2)	integer	Standard position centerline reflector

Table E-36. TZC, TZEC Local Common (CPU1)

Mnemonic	Type	Content
TZCSYL	integer array(2)	Syllabus file CHSAV
TZCSR	integer array(2)	Student record file CHSAV

Table E-37. VOCIN, Voice Data Collection Information

Mnemonic	Type	Content
VCFRAZ	integer array	Array (1:2, 1:N) in phrase id order Array (1, 1:N) is # of IFPs ready (maximum is 4 or 10) Array (2, 1:N) is IFP slot to be used (between 1-4 or 1-10)
VCWGT	integer array	Array (1:10) initialized to number of times features must be set given number of repetitions (0,0,0,1,1,1,2,2,2,3)

Table E-38. VLID, Validation Variables

Mnemonic	Type	Content
VLPCT	integer	Validation percentage requested
VLSTF	logical	True = STIFLE requested by student
VLARGS	integer array	Array (1:6). Phrases to be validated
VLTOC	integer	User clock counter
VLRES	integer array	Array (1:6). Resolution masks for recognition during validation phase. All masks are data initialized to zero.

APPENDIX F

File Structures

Syllabus File

The syllabus file guides the entire sequence of GCA-CTS training. It is a card image file containing the names of the individual phase instruction files, thus there is an entry for each phase of each task as well as for the performance test. Since not all phases are included in each task, the phase to which each file relates is identified. Position within this file is maintained by storing CHSAV information in the student file. File format is:

Column	Content	Meaning
1	C	Comment
	1	Phase 1 instruction file
	2	Phase 2 instruction file
	3	Phase 3 instruction file
	4	P run file
3-15	filename	Name of instruction file

Example:

```
C TASK  1.1
1 T1$1.01
C TASK  1.2
1 T1$2.01
2 T1$2.02
```

Phase 1 Instruction File

Phase 1 teaches the proper use of radio terminology while formulating the student's voice reference patterns. Creative utilization of the terminal display, graphics display, student panel, digitized audio, and voice synthesizer provide a variety of audio-visual aids to implement the phase 1 tasks. For each task a card image file defines the student voice data collection and instruction sequence. The file name identifies both task and phase, e.g. T1$3.01 is the task 1.3, phase 1 file. Eight types of cards may be used: comments, voice data collection instructions, display instructions, prompts, radar simulation instructions, aircraft simulation instructions, wait for events instructions, and task file sequence instructions. The seven latter card types are further divided into function card types. An instruction card generally contains the instruction type code in column 1 and the function identifier right-justified in columns 3 and 4, followed by the appropriate function arguments. Detailed format descriptions of each card type follow.

Comments

Comment cards are defined as follows:

Column	Content	Meaning
1	C	Comment
3-62	comment text	
63	x	End of record

Voice Data Collection Instructions

Voice Data Collection (VDC) incorporates collection of input feature patterns (IFPs), formulation of voice reference patterns (VRPs) for reference array storage, and validation of collected VRPs (matching VRPs to student inputs). IFP collection is limited to the storage of the ten most recently obtained patterns. Upon request, these IFPs are used to formulate VRPs which are stored in the reference array for the speech recognition base. Provisions are made for six VDC functions.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
VDC = 1	1		Start VDC. Must be the first VDC function requested.
1	2	Column 6-7 = xx, Phrase # 9-10 = xx 12-13 = xx . . . . 21-22 = xx	Collect IFP for phrase(s). Does not provide a prompt. Waits for phrase(s) input.
1	3	Column 6-7 = xx, phrase #	Form VRP for phrase. Necessary IFPs must be collected before this function is made valid.
1	4	Column 6-7 = xx% 9-10 = phrase # 12-13 = xx 15-16 = xx . . . . . . 24-25 = xx	Validate to % accuracy, # phrases. Issues a prompt and validates student response. Continues until required accuracy is attained. Prompt is issued by most recent prompting device.
1	5		Validate, no prompting. Student may voice any phrase(s) learned thus far. The recognized phrase(s) are echoed.
1	6		Terminate VDC. All VRPs for which IFPs have been collected must be completed before this function is made valid.

Display Instructions

The graphics display is controlled and directed by the five functions defined in the table below. Graphic images, referred to as pictures, may contain a particular symbol, text segment, background display, etc. All active pictures create the final display image.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Display = 2	1	Column 6-8 = xxx, picture #	Start picture. Activates requested picture for display.
2	2	Column 6-8 = xxx, picture #	Turn off picture.
2	3		Turn on display processor.
2	4		Turn off display. Turn off all active pictures.
2	5	Column 6-8 = xxx, picture #	Blink picture.

Prompts

Prompts may be issued via the CRT terminal display, the digitized audio, and the Votrax voice synthesizer. The model controller also may be called upon to give controller message prompts. A freeze after a particular message may be selected by setting columns 6-7 of the aircraft simulation instruction to the message number.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Prompts = 3	1	Column 6-8 = xxx, message #	Type message on CRT.
3	2	Column 6-8 = xxx, phase #	Transfer phrase to $VRO (Votrax).
3	3	Column 6-8 = xxx, phase #	Transfer digitized phrase to audio output device.
3	4	Column 6 = x, prompt device # 1-Votrax 2-CRT 3-Audio 8=x, phrase set # 1-all controller messages 2-messages for which VRPs have been defined 3-list follows: If option 3 has been selected for column 8: Column 10-11=xx, # of messages 13-14 = xx, list of messages 16-17 = xx 19-20 = xx . . . . 49-50 = xx	Activate model controller activity using specified device(s) for output and activating specified messages.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
3	5		Terminate model controller activity.
3	6	Column 6-8 = xxx, phrase #	Store audio input for given phrase.
3	7	Column 6-8 = xxx, message #	Type message on instructor CRT.
3	8	Column 6-7 = xx, student panel status	Change student panel status to requested status.

Radar Simulation Instructions

Azimuth and elevation servo alignment and position are the primary targets for the radar simulation instructions. Azimuth and elevation servo angles are expressed as zones with respect to the glideslope and extended runway centerline. The azimuth servo angle zones range from -2 to 2 with 0 representing the alignment of the azimuth servo with the glideslope. The angle the glideslope forms with the ground is bisected to produce the zones -1 and -2. Likewise, zones 1 and 2 are produced by bisecting the angle made by the glideslope and the upper limit of the azimuth servo angle. Elevation servo angles are similarly defined with the angle formed by the extended runway centerline and azimuth normal line being zone -1, alignment with the runway parallel line being zone 0, and the remaining 15° of azimuth sweep angle bisected being zones 1 and 2. The instruction cards are defined as follows:

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
Radar = 4	1	Column 6-7 = $\pm$ x	Activate azimuth servo. Initialize azimuth to zone x, $-2 \leq X \leq 2$.
4	2	Column 6-7 = $\pm$ x	Activate elevation servo. Initialize elevation to zone x, $-1 \leq x \leq 2$.
4	3	Column 6-7 = $\pm$ x	Turn off azimuth servo, freezing in zone x.*
4	4	Column 6-7 = $\pm$ x	Turn off elevation servo, freezing in zone x.*
4	5	Column 6 = x, $0 \leq x \leq 3$	Set azimuth alignment as requested; 0 = aligned, 3 = badly in need of alignment.
4	6	Column 6 = x, $0 \leq x \leq 3$	Set elevation alignment as requested; same criteria as above.
4	7	Column 6 = x, $0 \leq x \leq 3$	Set range alignment as requested; same criteria as above.

* If servo was not activated when this command is given, the system nonetheless moves the appropriate hash marks to the indicated zone.

Aircraft Simulation Instruction

Aircraft, pilot, and environment simulation variables are initialized and activated by this group of instructions. The environmental simulation set is a set of two cards which initialize all flight variables.

Card 1:

Column	Content	Meaning
1	5	Aircraft simulation instruction set.
4	1	First card of an environmental simulation set.
7		A/C type.
9-12	xx.x	Starting range from touchdown in miles.
14-17	xx.x	Ending range in radar miles.
19-21		Initial heading.
23		Indicates the form of the information in columns 25-33.
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 39-43.
39-40		Smaller valued zone (e.g., -2)
42-43		Larger valued zone (e.g., 1)
45	T,F	Azimuth target return suppression if T.
47	T,F	Elevation target return suppression if T.

Card 2:

Column	Content	Meaning
1	*	Continuation card
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch and go
	4	Minimum fuel
	5	No gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Waveoff
	5	Clearance given then cancelled.
		Wind Information:
7-9		Heading
11-12		Speed in knots
14-15		% Fluctuation.
17	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
19	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 21-24.
21-24	xx.x	Range from touchdown in miles when gyro failure occurs.
26-29	xx.x	Minimum separation (not used).
31	T,F	If T, cause icing to occur during the run.
33-36	xx.x	Range from touchdown when icing occurs.
38	T,F	If T, aircraft is to have hydraulic failure.
40	T,F	If T, aircraft is to have single engine failure.

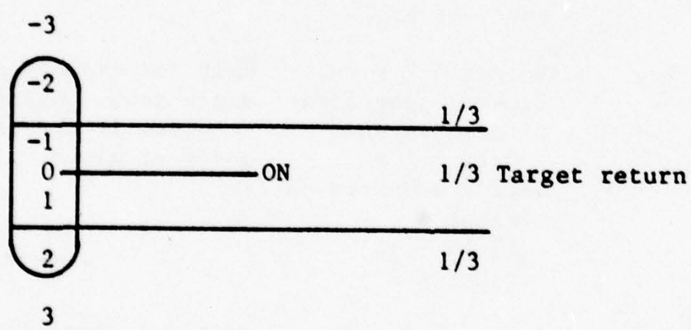
Aircraft dynamics are switched on and off by the instruction card which follows:

Column	Content	Meaning
1	5	Aircraft simulation instruction set.
4	2	Begin A/C dynamics.
6-7	XX	Freeze A/C dynamics on event XX.

The freeze occurs after the controller advisory has been issued. This card type may be used several times during a given simulation to freeze then to continue and freeze on another event. If no freeze event number is defined, the simulation continues until the aircraft reaches the ending range given in the environmental simulation set.

Wait for Events

The events described in this section generate a wait condition. One of these event types, wait for aircraft azimuth/elevation zone uses the zone definitions shown here:



The zone -4 is used to indicate a low altitude alert for the aircraft elevation zone. The card formats are as follows:

Instruction Type (Column 1)	Identifier Column	Arguments	Function
Wait = 6	1	Column 6-7 = xx seconds	Delay xx seconds before next operation.
6	2	Column 6-7 = xx, time-out specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout, 13-14=xx_1 16-17=xx, skip # of cards on entry of xx_1 . . . 43-44 = xx_6 46-47 = xx, skip # of cards on entry of xx_6	Wait for keyboard entries entries matching xx_1, xx_2, ...xx_6.
6	3	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout 13-14 = $\pm$X $-2 \leq X \leq 2$	Wait for azimuth servo angle zone. Notification is given if the servo is moved at all.*
6	4	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout 13-14 = $\pm$ X, $-1 \leq X \leq 2$	Wait for elevation servo angle zone. Notification is given if the servo is moved at all.*
6	5	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout 13-14 = $\pm$ X, $-3 \leq X \leq 3$	Wait for aircraft azimuth zone.

Instruction Type (Column 1)	Identifier Column	Arguments	Function
6	6	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout 13-14 = $\pm$ X, $-3 \leq X \leq 3$	Wait for aircraft elevation zone.
6	7	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout 13-14 = x.x	Wait for aircraft range from touchdown, x.x in miles.
6	8	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout	Wait for Votrax to finish speaking.
6	9	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx skip # of cards on timeout 13-14 = xx	Wait for student panel input.
6	10	Column 6-7 = xx, timeout specification in seconds 9-11 = $\pm$ xx, skip # of cards on timeout	Wait for any student voice input (waits one second after end of input).

* Flag 10 is set to 99 upon servo movement. Refer to sequence instructions for discussion on flags and the conditional statement.

Task File Sequence Commands

Normal sequencing through the phase 1 task file consists of a sequential card by card process with the exception of skip on timeouts and key entry discussed in the previous section. The processing sequence may be altered by utilization of skips, subroutine calls and returns, and conditional skips provided by the command formats listed below.

Instruction Type (Column 1)	Identifier (Column 4)	Arguments	Function
7	1	Column 6-8 = $\pm$ xx	Skip # of cards.
7	2	Column 6-7 = xx, entry record # 9-11 = $\pm$ xx_1 . . . 25-27 = $\pm$ xx_5, skip # of cards from normal return point	Subroutine call with provisions for abnormal returns. Five levels of nesting are allowed.
7	3	Column 7 = x, subroutine return # 0 - normal return 1 - return to start of task file 2,3...,6 - return to corresponding abnormal return point specified in the subroutine call	Subroutine return.
7	4	Column 6-7 = xx, flag # (1-10) 9-10 = xx, condition #	Set flag to condition number.
7	5	Column 6-7 = xx, flag # (1-10) 9-10 = xx, condition # 12-14 = $\pm$ xx, skip xx records	If flag is set to the given condition #, skip is performed.

Phase 2 Problem Specification File

In phase 2, the system freezes and gives feedback to the student whenever an error is made on the new material. One phase 2 instruction file exists for every task which provides phase 2 training. It is a card image file whose suggested name incorporates both task and phase of instruction, e.g., for task 3.2 the file name should be T3$2.02. The file contains three types of cards: comments, the header information card, and environmental simulation sets (two cards each). The file contains exactly one header information card, and this preceeds any environmental simulation sets. In general, the header information pertains to all problems while the environmental simulation sets describe a particular problem. Card formats are described in the tabular form below. The code in column one specifies the card type as follows:

Column	Content	Meaning
1	C	Comment*
	1	First card of header information set. One such set is required for each file. It must preceed the first environmental simulation set.
	2	First card of environmental simulation set.
	*	Second or continuation card of environmental set.

*All comments prior to header card will be displayed on the instructor's CRT.

Header Information

The header consists of one card whose format is as follows:

Column	Content	Meaning
1	1	Identifies this as header.
3		Azimuth radar display:
	0	off
	1	on
	2	on, no hash marks.
5	T,F	Azimuth servo on if T, else off.
7		Elevation radar display
	0	off
	1	on
	2	on, no hash marks.
9	T,F	Elevation servo.
11-17		Text file name (e.g., T3$2.P2) text to be displayed on student's CRT, or blank.
		Performance variables which will cause the system to freeze when an error occurs:
19-20		PMV number
22-23		PMV number
.		.
.		.
.		.
	-1	Indicates the end of freeze PMVs.

Environmental Simulation Card Sets

The environmental simulation is given in two-card sets described below.

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3		Number of consecutive error free repeats of this problem.
7		A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.
19-21		Initial heading.
23		Indicates the form of the information in columns 25-33:
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 39-43.
39-40		Smaller valued zone (e.g., -2).
42-43		Larger valued zone (e.g. 1).
45	T,F	True if handoff to be given, else false.

Card 2:

Column	Content	Meaning
1	*	Continuation card
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch and go
	4	Minimum fuel
	5	No gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Waveoff
	5	Clearance given then cancelled
		Wind Information:
7-9		Heading
11-12		Speed in knots
14-15		% Fluctuation.
17	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
19	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 21-24.
21-24	xx.x	Range from touchdown in miles when gyro failure occurs.
26-29	xx.x	Minimum separation.
31	T,F	If T, cause icing to be encountered at the point specified in columns 33-36.
33-36	xx.x	Range in miles when icing commences.
38	T,F	If T, hydraulic failure at initialization.
40	T,F	If T, single engine failure at initialization.

Phase 3 Problem Specification File

Phase 3 problems are scored exercises which allow the student to practice and integrate his new skills. One phase 3 instruction file exists for every task which provides phase 3 training. It provides for both individual problems and randomly selected problems, with scoring of specified performance measurement variables. It is a card image file whose suggested name incorporates both task and phase of instruction, e.g., for task 3.2 the file name should be T3$2.03. The file contains three types of cards: comments, the header information set (two cards) and environmental simulation set (two cards each). The file contains exactly one header information set, and this preceeds all environmental simulation sets. In general, the header information pertains to all problems and includes information such as PMVs to be scored, etc. The environmental simulation sets describe a particular problem or set of problems. Note that a sufficient number of exercises is included to provide the maximum number of runs specified. A simple way to ensure this is to specify a multi-possibility exercise as the last one in the file.

Card formats are described in tabular form below. The code in column one specifies the card type as follows:

Column	Content	Meaning
1	C	Comment*.
	1	First card of header information set. One header information set is required for each phase 3 file; it must preceed the first environmental simulation set.
	2	First card of environmental simulation set.
	*	Second or continuation card of header or environmental set.

*All comments prior to header will be displayed on the instructor's CRT.

Header Information

The header consists of two cards of information. The first is similar to that for phase 2. The second contains the error scores which must not be exceeded in order to pass at this level. These scores are integer percentage error scores which will be compared to the average error score over the last ten problems or over the minimum number of runs, whichever is smaller. The format of the cards is as follows:

Card 1:

Column	Content	Meaning
1	1	Identifies this as header information.
3		Azimuth radar display:
	0	off
	1	on
	2	on, no hash marks.
5	T,F	Azimuth servo, on if T, else off.
7		Elevation radar display:
	0	off
	1	on
	2	on, no hash marks.
9	T,F	Elevation servo.
11-12		Minimum number of runs.
14-15		Maximum number of runs.
17-23		Text file name, text to be displayed on student CRT; or blank.
25-26		PMV number relating to this skill (score checked first).
28-29		PMV number relating to this skill.
.		.
.		.
.		.
	-1	End of PMV numbers relating to this skill.

Card 2:

Column	Content	Meaning
1	*	Identifies this card as the continuation card.
3-5		Maximum allowable % error score to pass PMV1, or -1 if not scored.
7-9		Maximum allowable % error score to pass PMV2, or -1.
.		.
.		.
.		.

Environmental Simulation Card Sets

The environmental simulation information is given in two-card sets. These sets are of two types, distinguished by a code in column 3. The first type provides initialization information for one run (which may be repeated). The second type provides a range of information about environmental parameters from which individual problem parameters are chosen randomly.

Single Problem Specification

This card set provides information regarding one run, which may be repeated. Note that if the repeat feature is chosen, no further cards will be examined in this file, rather the system will continue to use this information for problem setup until the conditions for progressing to the next task are met. The format of the cards in this set is shown below. These cards differ from the phase 2 environmental simulation cards only in columns 3 and 5 of the first card.

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3	1	The information is for one run.
5	T,F	T if this same problem is to be repeated, else F. If T, no more environmental card sets may follow.
7		A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.
19-21		Initial heading.
23		Indicates the form of the information in columns 25-33.
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left, or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller advisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 39-43.
39-40		Smaller valued zone (e.g., -2).
42-43		Larger valued zone (e.g. 1).
45	T,F	True if handoff to be given, else false.

Card 2:

Column	Content	Meaning
1	*	Continuation card.
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch and go
	4	Minimum fuel
	5	No gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Waveoff
	5	Clearance given then cancelled.
		Wind Information:
7-9		Heading
11-12		Speed in knots
14-15		% Fluctuation.
17	T,F	If T, cause A/C to descend to a point which requires a low altitude alert be given once during the run.
19	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 21-24.
21-24	xx.x	Range from touchdown in miles when gyro failure occurs.
26-29	xx.x	Minimum separation.
31	T,F	If T, cause icing to be encountered at the point specified in columns 33-36.
33-36	xx.x	Range in miles when icing commences.
38	T,F	If T, hydraulic failure at initialization.
40	T,F	If T, single engine failure at initialization.

Multi-Possibility Problem Specification

This card set provides a range of initial conditions for problem set up. The phase 3 executive selects individual run parameters randomly from those specified, and thus can provide a variety of similiar problems as required for adaptive training. Note that, as with the repeat option on a single problem specification, no further cards are examined since the system continues to use this information for problem setup until the conditions for progressing to the next task are met. The format of the cards in this set is shown below.

Card 1:

Column	Content	Meaning
1	2	First card of an environmental simulation set.
3	2	Multi-possibility problem parameters.
		Starting positions. Percentages in columns 5-31 must sum to 100. Handoffs always given.
5-7		% of minimum fuel problems starting in position 1 (right base, minimum fuel, 4 miles, 1000 ft, heading 140° ± variation, clearance always given).*
9-10		Maximum heading variation from 140° on position 1 starts. ($\lvert var \rvert \leq 10°$).
12-14		% of problems starting in position 2 (right base, 8 miles, 1500 ft, heading 140° ± variation).*
16-17		Maximum heading variation on position 2 starts. (Again, $\lvert var \rvert \leq 10°$).
19-21		% of problems starting in position 3 (straight in approach, 11 miles, 1500 ft. heading 160° ± variation).*
23-24		Maximum heading variation on position 3 starts.
26-28		% of problems starting in position 4 (left base, 8 miles, 1500 ft, heading 180° ± variation.)*
30-31		Maximum heading variation on position 4 starts.
		Aircraft speeds. Sum of values in columns 33-43 must equal 100.
33-35		% slow A/C.
37-39		% medium A/C.
41-43		% fast A/C

*Note: these are track headings. The simulated pattern controller will supply appropriate crab to the assigned heading to produce the specified track heading.

Card 1:

Column	Content	Meaning
		Pilot type. Sum of values in columns 45-63 must be 100.
45-47		% type 1 (best).
49-51		% type 2.
53-55		% type 3.
57-59		% type 4.
61-63		% type 5 (worst).

Card 2:

Column	Content	Meaning
1	*	Continuation card.
		Approach type. Sum of values in columns 3-13 must be 100. Applies only for starting positions 2-4. Minimum fuel is full stop.
3-5		% full stop.
7-9		% low approach.
11-13		% touch and go.
		% no gyro approaches. These gyro failures will be uniformly distributed between 3 and 5 miles. If the failure occurs at 5+ miles (or 3+ miles on minimum fuel approaches), it will be announced by the pattern controller in the handoff message.
		Clearance information for tower simulation. Note: Clearance will always be given on minimum fuel approaches. These values apply only to other types of approaches. Sum of values in columns 23-41 must be 100.
19-21		% clearance given at first request.
23-25		% continue, then clear at 2 miles.
27-29		% not given (no response).
31-33		% waveoff
35-37		% clearance given then cancelled.
		Wind conditions. Sum of values in columns 43-61 t must be 100.
39-41		% light and variable.
43-45		% 190° at 10 kts.
47-49		% 190° at 20 kts.
51-53		% 250° at 10 kts.
55-57		% 250° at 20 kts.

Card 2:

Column	Content	Meaning
59-61		% of runs in which A/C is forced to descend to a point which requires a low altitude alert.
63-65		% of problems in which minimum separation is not maintained.
67-69		% of problems in which icing is encountered between 1-5 miles.
71-73		% of problems in which hydraulic failure occurs at initialization.
75-77		% of problems in which single engine failure occurs at initialization.

AD-A069 036 LOGICON INC SAN DIEGO CA TACTICAL AND TRAINING SYSTE--ETC F/G 5/9
GROUND CONTROLLED APPROACH CONTROLLER TRAINING SYSTEM.(U)
APR 79 G D BARBER, M HICKLIN, C MEYN N61339-77-C-0162
UNCLASSIFIED 5581-0005 NAVTRAEQUIPC-77--C-0162-2 NL

AD
A069036

END
DATE
FILMED
7-79
DDC

Performance Run Specification File

The performance run (P-run) will be used to determine whether or not the student passes the course. Because of its importance, final scoring will be deferred until the trainee and the instructor have the opportunity to review the replay of the run and correct any misrecognitions. The file consists of three types of cards: comments, the header card, and an environmental simulation card set, distinguished by a code in column 1 as follows:

Column	Content	Meaning
1	C	Comment*.
	1	Header information.
	2	First card of environmental simulation set.
	*	Second or continuation card of environmental set.

*Comments prior to the header card will be displayed on the instructor's CRT.

Header Information

The header consists of one card whose format is as follows:

Column	Content	Meaning
1	1	Identifies this as the header.
3-5		Maximum allowable error score to pass PMV1.
7-9		Maximum allowable error score to pass PMV2.
.		.
.		.
.		.

Environmental Simulation Card Set

One environmental simulation card set must be provided to describe the P-run. The set differs from the phase 2 environmental simulation cards only in columns 3 and 5 of the first card.

Column	Content	Meaning
1	2	First card of an environmental simulation set.
7	.	A/C type:
	1	U-21
	2	A6
	3	P3
	4	T38.
9-12	xx.x	Starting range from touchdown in radar miles.
14-17	xx.x	Ending range in miles.
19-21		Initial heading.
23		Indicates the form of the information in columns 25-33.
	1	A/C starting position specified in relation to environmental simulation, i.e., in feet and miles.
	2	A/C starting position specified in relation to display simulation, i.e., by zone.
25-28		A/C starting altitude in feet or by zone (right justified).
30-33		A/C starting offset in miles (xx.x) negative if left or by zone (right justified).
35		Pilot type: 1 = best, 5 = worst.
37		Type of flight:
	1	Pilot responds normally to controller adivisories.
	2	Restrict A/C position to contiguous azimuth zones given in columns 39-43.
	3	Restrict A/C position to contiguous elevation zones given in columns 39-43.
39-40		Smaller valued zone (e.g., -2).
42-43		Larger valued zone (e.g., 1).
45	T,F	True if handoff to be given, else false.

Card 2:

Column	Content	Meaning
1	*	Continuation card
3		Approach type, used for the handoff message as follows:
	1	Full stop
	2	Low approach
	3	Touch and go
	4	Minimum fuel
	5	No gyro.
5		Clearance information for tower simulation:
	1	Clearance given at first request
	2	Continue, then clear at 2 miles
	3	Not given (no response)
	4	Wave off
	5	Clearance given then cancelled
		Wind information:
7-9		Heading
11-12		Speed in knots
14-15		% Fluctuation.
17	T,F	If T, cause A/C to descend to a point which requries a low altitude alert be given once during the run.
19	T,F	If T, cause a gyro failure after handoff, at the point specified in columns 21-24.
21-24	xx.x	Range from touchdown in miles when gyro failure occurs.
26-29	xx.x	Minimum separation.
31	T,F	If T, cause icing to be encountered at the point specified in columns 33-36.
33-36	xx.x	Range in miles when icing commences.
38	T,F	If T, hydraulic failure at initialization.
40	T,F	If T, single engine failure at initialization.

Demonstration Run Specification File

The demonstration runs that take place while the system is idle are chosen randomly from the set of posibilities provided in this file. The file consists of a two card set which is identical in format to the phase 3 multi- possibility environmental card set described previously.

Remedial Training File

Remedial training problems (other than rule explanations for knowledge items) are selected form this file. The file is a randomly organized ASCII text file with 8 word records which can be thought of as being blocked by level.

Record	Word	Content	Meaning
1-20	1	2△	Level 2 remedial training.
	2	01	PMV number.
	3		Phase of instruction, right justified.
	5-8		Problem file name or "NONE."
21-40	1	3△	Level 3 remedial training.
.	.		
.	.		
.	.		

Voice Data

Two voice data files are formulated and stored for each trainee. They are the IFP.VO and the VRP.VO files which hold input feature pattern and voice reference pattern information respectively. These files reside within the trainee's own subdirectory on the removable disk.

IFP.VO is composed of 32 word records. Records 1 through 4 contain the number of IFPs available for VRP formulation. This number may range from 0 (no IFP has been collected for the phrase) to 4 or 10 (maximum number of IFPs have been collected for the phrase). Records 5 through 8 hold pointers to the next IFP storage slot. The pointer references an empty storage slot or the oldest IFP in storage. The remaining records are allocated for actual IFP storage. Each phrase is represented by a maximum of 4 or 10 of the most recently collected IFPs. Distinct phrases require 4 IFPs for VRP formulation whereas less distinguishable phrases require 10 IFPs. All phrase IFPs and their references are stored in phrase identification order.

VRP.VO is also arranged in phrase identification order and 32 word records. VRP present flags comprise records 1 through 4 of VRP.VO. These flags are set upon phrase VRP formation. Records 5 through 8 store validation percentages for the VRPs which have been validated. Remaining records are given to VRP storage. The VRPs, like the IFPs are stored in 32 time slot (64-word) or 16 time slot (32 word) format as specified by syllabic length (phrases of three syllables or less are represented by 16 time slot VRPs). Filler blocks of one record length are utilized to maintain VRP starts at 1024-word boundaries for window mapping purposes.

Votrax Phrase File (FRAZ.VO)

FRAZ.VO is a contiguous file of octal phoneme codes which the Votrax uses to produce audible output. Each record consists of 64 octal words padded with "-1". The high order 2 bits of each octal word contain information on inflection and the bottom six contain phonemes. One GCA phrase is represented by each record. The file is created from a vocabulary file called VOCAB which contains each word to be used with its coded form. This file is translated into ASCII phoneme codes by a program called PHRASE, then to octal phonemes by the program CODER. The contiguous octal format ensures that speed of execution will be optimal.

Special Purpose Digitized Speech (SPSPH)

Individual phrase segments of digitized speech are stored in this file. It is a randomly organized file with 500 word records, corresponding to the size of the speech digitizer I/O buffers. A special index file into this file is maintained which gives the starting record and number of records associated with each phrase.

Prompting Digitized Speech File

This file and its corresponding index file are constructed in the same way that the special purpose digitized speech file and its index are. It is used to store the trainee's own speech for prompts, etc.

Speech Data Replay File (RPLSPH)

Speech data are digitized and saved for replay in this file. It is contiguously organized. Records are the same size as the input size and consist of speech data only. One record is written every .5 second throughout the run.

Radar Data Replay File (RPLDSP)

This file is written by RADAR and read during replay to recreate the radar display. It is a random file, with 8 words per record. Its contents are described below.

Word	Contents
1	X coordinate of target.
2	Y_1, lower coordinate of azimuth target.
3	Y_2, upper coordinate of azimuth target.
4	Y_3, lower coordinate of elevation target.
5	Y_4, upper coordinate of elevation target.
6	Y_A, azimuth servo position.
7	Y_E, elevation servo position.
8	unused.

Activity Replay File (RPLACT)

Display starting conditions, all recognition information, student panel inputs, servo inputs and synthesized speech outputs are stored in this file. It is randomly organized, with 8 words per record. There are six types of records which are distinguished by a code in column 1. These records are described below.

Record Type	Word	Content	Meaning
Header	1	1, -1	Identifies this as a header record. If -1, this is a P-run replay file.
	2		Initial DOA to student's panel.
	3		Initial DOB to student's panel.
	4	T,F	T if picture 1 is on.
	5	T,F	T if picture 2 is on.
	6	T,F	T if picture 3 is on.
	7	T,F	T if picture 4 is on.
	8	T,F	T if picture 7 is on.
Header, record 2	1	T,F	True if picture 8 is on.
	2	T,F	True if picture 9 is on.
	3	T,F	True if picture 10 is on.
	4	T,F	True if picture 11 is on.
	5		Initial servo X coordinate.
	6		Initial azimuth servo Y coordinate.
	7		Initial elevation servo Y coordinate.
	8	T,F	True if feeder gives handoff.

Record Type	Word	Content	Meaning
Model Controller Selections	1	2	Identifies this as a model controller record. It is always written immediately prior to a recognition record. It consists of two 8 word records.
	2	CTGPP	Correct glidepath position message.
	3	CTGPT	Correct glidepath trend message.
	4	CTCRP	Correct course position message.
	5	CTCRT	Correct course trend message.
	6	CTRNG	Correct range, including DH and OLT, or -1.
	7	CTOTHR	Other final controller messages.
	8	CTEMERG	Emergency messages.
	1		Azimuth zone.
	2		Azimuth trend.
	3		Elevation zone.
	4		Elevation trend.
	5-6		Miles to touchdown.
	7-8		Not used.

Record Type	Word	Content	Meaning
SUS Output	1	3	Identifies this as a recognition block.
	2		Time of LP_4 in .5 second ticks from the start of the problem.
	3		Time in 100 msec ticks from above timer.
	4		First choice message understood (message actually given to APE).
	5		Heading, etc., if any.
	6		Call sign, or wind speed.
	7		Second choice message understood, or -1.
	8	T,F	T if correction was applied to this phrase.

Record Type	Word	Content	Meaning
Panel Changes	1	4	Identifies this as a panel change.
	2	1,2	If 1, DIA and DIB; if 2, DOA and DOB.
	3		Time in .5 sec ticks.
	4		DIA word or DOA word.
	5		DIB word or DOB word.
	6-8		Not used.
Servo Net Changes	1	5	Identifies this as servo information.
	2		Time in .5 sec ticks.
	3		Servo X coordinate.
	4		Azimuth servo Y coordinate.
	5		Elevation servo Y coordinate.
	6-8		Not used.
Automated Voice Output	1	6	Identifies this as GLIB ouput information.
	2		Time in .5 sec ticks.
	3		First phrase to be output.
	.		
	.		
	.		
	n	-1	End of phrases to be output.
	n+1-8		Not used.

Record Type	Word	Content	Meaning
Special Scoring Timers, etc.	1	7	Identifies this as a special scoring parameter record.
	2	1	Identifies this as a time 50% of target appears type record indicator.
	3		Time 50% of target appears.
	4-8		Unused.
	1	7	As above.
	2	2	Identifies this as a mile mark record.
	3		Miles to touchdown.
	4		Distance from azimuth cursor in feet.
	5		Distance from elevation cursor in feet.
	6-8		Unused.
	1	7	As above.
	2	3	Identifies this as a OLT or ADH record.
	3	1,2	1:ADH; 2:OLT.
	4		Azimuth zone.
	5		Elevation zone.
	6-8		Unused.

Student Performance File

PMV data are saved in this file for every problem undertaken. It is randomly organized, with 80 words per record. The first record contains header information (the student's name, record number of performance test data, next problem, etc.). Subsequent records contain task, problem, and PMV measures PV01-PV23, as well as indicators showing which problems were ADAPTed due to poor performance, which phase 2 problems were skipped, and other performance information.

APPENDIX G

String Charts

The following definitions describe string chart construction:

a. Event: An event definition is typed on the left-hand side of the page and is symbolized by a handdrawn horizontal line.

b. Path: A handdrawn vertical line denotes a path between events.

c. Entrance point: An entrance to an event is at an intersection of a vertical and horizontal line where a handdrawn bullet exists. There is one bullet for each entrance. A vertical line is associated with each entrance. There may be one or several entrance points per event.

d. Exit point: An exit from an event is at an intersection of a vertical and horizontal line where no bullet exists, or where handdrawn alphanumeric character(s) exist. Where decisions or a variety of exits exist, they may be labeled with one to several alphanumeric characters.

e. Special exit/entrance points: When it is awkward to draw a continuous solid line between events, vertical lines may be used in conjunction with handdrawn connectors. A connector shall consist of the entrance point page number and a unique lower case letter defining the particular entrance point.

f. Crossing: Where a path passes through an event in a straight line neither an exit nor an entrance is implied.

g. Loop: Loop entries are distinguished by parentheses. All references to the entry are likewise parenthesized.

DISTRIBUTION LIST

Naval Training Equipment Center Orlando, FL 32813	18
Defense Documentation Center Cameron Station Alexandria, VA 22310	12